RILSON MACHADO DE OLIVEIRA

PREDIÇÃO DE ESTRUTURAS SECUNDÁRIAS DE PROTEÍNAS UTILIZANDO REDES NEURAIS ARTIFICIAIS

Monografia de graduação apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras como parte das exigências do curso de Ciência da Computação para obtenção do título de Bacharel em Ciência da Computação.

LAVRAS
MINAS GERAIS – BRASIL
2008

RILSON MACHADO DE OLIVEIRA

PREDIÇÃO DE ESTRUTURAS SECUNDÁRIAS DE PROTEÍNAS UTILIZANDO REDES NEURAIS ARTIFICIAIS

Monografia de graduação apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras como parte das exigências do curso de Ciência da Computação para obtenção do título de Bacharel em Ciência da Computação.

Área de Concentração:

Bioinformática

Orientador

Thiago de Souza Rodrigues

LAVRAS
MINAS GERAIS – BRASIL
2008

Ficha Catalográfica

Oliveira, Rilson Machado

PREDIÇÃO DE ESTRUTURAS SECUNDÁRIAS DE PROTEÍNAS UTILIZANDO REDES NEURAIS ARTIFICIAIS / Rilson Machado de Oliveira. Lavras – Minas Gerais, 2008. 59p : il.

Monografia de Graduação – Universidade Federal de Lavras. Departamento de Ciência da Computação.1. Bioinformática. 2. Redes Neurais Artificiais. I. CIPRIANI, O. N. II. Universidade Federal de Lavras. III. Título.

RILSON MACHADO DE OLIVEIRA

PREDIÇÃO DE ESTRUTURAS SECUNDÁRIAS DE PROTEÍNAS UTILIZANDO REDES NEURAIS ARTIFICIAIS

Monografia de graduação apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras como parte das exigências do curso de Ciência da Computação para obtenção do título de Bacharel em Ciência da Computação.

Aprovada em

_____
Cláudio Fabiano Motta Toledo

_____
Cristiano Leite de Castro

_____
Thiago de Souza Rodrigues
(Orientador)

LAVRAS
MINAS GERAIS – BRASIL

PREDIÇÃO DE ESTRUTURAS SECUNDÁRIAS DE PROTEÍNAS UTILIZANDO REDES NEURAIS ARTIFICIAIS

RESUMO

A pesquisa se encontra na área de bioinformática, objetiva-se a predizer a estrutura secundária de uma proteína a partir de sua seqüência de aminoácidos, ou seja, sua estrutura primária. A predição foi feita utilizando redes neurais artificiais, que é um modelo computacional baseado no funcionamento de neurônios. Um banco de dados de proteínas, PDB (Protein Data Bank) foi utilizado para obter as informações das seqüências. Ao fim da pesquisa obteve uma taxa de exatidão de 78.1 % para a predição

Palavras-chave: Bioinformática, Redes Neurais Artificiais.

Prediction of secondary structures proteins with neural network

ABSTRACT

This bioinformatics research aims the prediction of protein secondary structures from its amino acid sequence, in other words, its primary structure. The prediction will be accomplished using artificial neural networks, which are a computational model based on the behavior of neural cells. A protein database PDB (Protein Data Bank) will be used in order to obtain information on the sequences.In the end this research the accuracy was 78.1%.

Keywords: bioinformatics, artificial neural networks, protein structure prediction.

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

1 INTRODUÇÃO

Nesta seção será dada uma pequena contextualização do presente trabalho e suas motivações. Logo em seguida, também são descritos seus principais objetivos.

1.1 Contextualização

As proteínas estão presentes em todos os organismos vivos, elas desempenham um papel fundamental nestes organismos, sendo uma estrutura básica e fundamental para a vida. Esses componentes básicos desempenham funções variadas, ter o conhecimento da função realizada pelas inúmeras proteínas é de grande utilidade, pois com essas informações podem-se diagnosticar doenças, descobrir curas, desenvolver novos medicamentos, entre outras inúmeras utilidades.

As proteínas são formadas por partes menores, denominadas aminoácidos. Uma seqüência de aminoácidos constitui uma proteína, e através dessa seqüência pode-se inferir sobre a função da proteína. A função de uma proteína é determinada por sua estrutura tridimensional, que é determinada pela natureza e seqüência de seus aminoácidos. (KREUZER & MASSEY, 2002)

Ainda de acordo com Kreuzer & Massey (2002), a estrutura tridimensional de centenas de proteínas tem sido minuciosamente determinada com a utilização de técnicas de cristalografia de raios X e ressonância magnética nuclear. Essas técnicas de se determinar as estruturas ainda são muito caras.

Por outro lado, a determinação da seqüência de aminoácidos de uma proteína em laboratório é fácil e, relativamente, pouco dispendiosa. No entanto, os procedimentos existentes para a determinação da estrutura tridimensional a partir da seqüência de aminoácidos não tem produzido resultados considerados adequados.

O termo proteína deriva do grego *proteíos*, "que tem prioridade", "o mais importante". Elas são consideradas as macromoléculas mais importantes das células. E para muitos organismos, constituem quase 50% de suas massas.

De acordo com Fonseca (2001) as proteínas são compostos orgânicos de estrutura complexa e massa molecular elevada, que vai de 5.000 a 1.000.000 ou mais unidades de massa atômica, sintetizadas pelos organismos vivos através da condensação de um grande número de

moléculas de alfa-aminoácido, através de ligações denominadas ligações peptídicas. Uma proteína é um conjunto de 100 ou mais aminoácidos, sendo os conjuntos menores denominados Polipeptídios.

Para Branden & Tooze (1991) proteínas são biopolímeros que possuem como alfabeto um conjunto de 20 aminoácidos, as proteínas são responsáveis por várias funções nos organismos vivos, dentre elas, ações de catálise. Um dos mais conhecidos e importantes tipos de proteínas são as enzimas, que exercem um papel muito importante nos organismos vivos.

A partir da seqüência de organização dos 20 aminoácidos é que a função se destaca. A combinação destes aminoácidos possibilita 10^{11} ou mais possíveis seqüências de aminoácidos, ou melhor, proteínas. Dos 20 aminoácidos, alguns são essenciais, ou seja, não são produzidos pelos organismos. Para o homem, 10 são essenciais(valina, leucina, isoleucina fenilalanina, triptofano, treonina, lisina, arginina, histidina e metionina).

E de acordo com Dill (1990) as proteínas são polímeros de unidade monoméricos, denominadas aminoácidos, contendo um grupo amina (NH2), um grupo ácido carboxílico (-COOH) e um radical R.

Para Lehninger (1984) as proteínas são formadas a partir da união de muitos aminoácidos e elas possuem diversas funções nos mais diversos organismos. A partir disso, pode-se notar que as proteínas não são somente as mais abundantes macromoléculas, mas também, são muito importantes para a vida. As milhares de enzimas que um organismo possui são todas proteínas com funções importantes. As informações genéticas, por exemplo, são expressas através de proteínas. Se classificar pela função uma proteína poderá ser:

- Enzimas: Proteínas altamente especializadas e com atividade catalítica. Existem mais de 2000 enzimas conhecidas, cada uma capaz de catalisar um tipo diferente de reação química.

- Proteínas transportadoras: São as responsáveis por transportar especificadamente moléculas ou íons de um órgão para outro. Um exemplo é a hemoglobina, responsável pelo transporte de oxigênio dos pulmões aos outros órgãos e tecidos.

- Proteínas Contráteis ou de movimento: São elas as responsáveis pela função de contração de algumas células. Também, são as responsáveis pela mudança de forma e

movimento de algumas células. Exemplos deste tipo de proteína são a actina e a miosina, que estão presentes no sistema contrátil de músculos esqueléticos.

- Proteínas Estruturais: São proteínas que servem para dar firmeza e proteção à organismos. Um exemplo muito comum deste tipo de proteína é o colágeno, altamente encontrado em cartilagem e tendões, sendo bastante resistente à tensão. Unhas e cabelos são formados, basicamente, por queratina, um outro tipo de proteína estrutural.

- Proteínas de defesa: São proteínas com função de defesa de organismos contra invasões de outras espécies. Exemplo disso, são os leucócitos (glóbulos brancos, anticorpos), proteínas especializadas com função de reconhecer e neutralizar vírus, bactérias e outras proteínas estranhas. Fibrinogênio e trombina são outras proteínas responsáveis pela coagulação do sangue e prevenção de perda sanguínea em casos de cortes e machucados.

Ao se falar em proteínas uma definição muito importante é a dos aminoácidos.

Para Copelland (1993) aminoácidos são compostos orgânicos que possuem uma estrutura básica comum, consiste de um carbono central denominado carbono α, o qual possui quatro ligantes diferentes, um grupo carboxila (COOH), um grupo amino (HH2) e um radical R também chamado cadeia lateral do aminoácido, que pode consistir desde um único átomo de hidrogênio até complexos anéis aromáticos, conforme podemos ver na Figura 1.1.

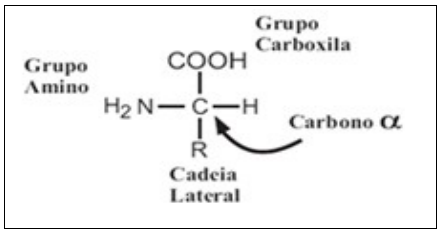

Figura 1.1: Estrutura do aminoácido Fonte: dados do trabalho

As proteínas são formadas a partir de um conjunto de vinte aminoácidos que se diferenciam pelas suas cadeias laterais. Quando presentes em proteínas, os aminoácidos são

denominados de resíduos, pois no processo de formação de proteína ocorre a perda de átomos, geralmente uma molécula de água – H_2O, que compunha a estrutura completa do aminoácido.

Segundo Petsko & Ringe (2004) os aminoácidos podem ser divididos em três diferentes classes, dependendo da natureza química da cadeia lateral. A primeira classe compreende os aminoácidos com cadeia lateral estritamente hidrofóbica, isto é o composto da cadeia lateral não se dissolve em contato com a água, alguns exemplos Alanina, Valina, Leucina, Isoleucina, Fenilalanina e Prolina. Aminoácidos que possuem cadeia lateral estritamente hidrofóbica, isto é, o composto da cadeia lateral se dissolve em contato com a água, compõem a segunda classe, exemplos de aminoácidos hidrofóbicos, Ácido Aspártico, Ácido Glutâmico, Serina, Treonima, Cisteína, Asparagina, Glutamina, Histidina e Argenina. A terceira classe é composta pelos aminoácidos com características polares e apolares, também chamados anfipáticos, que são os Lisina, Tirosina, Metionina e Triptofano. Tanto na segunda classe como na terceira classe os aminoácidos desovem na presença de água.

Quanto a ionização os aminoácidos são substâncias anfóteras, ou seja, pode atuar como ácidos ou como bases. Existem 2 grupos ácidos fortes ionizados, um –COOH e um –NH3+ . Em solução essas duas formas estão em equilíbrio protônico. R-COOH e R-NH3+, representam a forma protonada ou ácida, parceiras nesse equilíbrio. E as formas R-COO- e R-NH2 são as bases conjugadas. Assim, dependendo do meio, os aminoácidos podem atuar como ácidos (protonado, podendo doar prótons), neutros (a forma protonada e a forma receptora de prótons em equilíbrio) e base (base conjugada do ácido correspondente, ou seja, perdeu prótons, e agora é receptora deles). Os aminoácidos reagem com o ácido nitroso produzindo nitrogênio e um hidroxi-acido. A aplicação desta reação é a determinação da dosagem de aminoácidos,no sangue, medindo-se o volume de nitrogênio produzido (método de Slyke). Na putrefação dos organismos, certas enzimas reduzem os aminoácidos em aminas como a putrescina e a cadaverina.(PETSKO & RINGE, 2004)

O nome, a simbologia, a abreviação a e a nomenclatura dos 20 tipos de diferentes de aminoácidos podem ser observados na Tabela 1.1 e a Figura 1.2 expõe as suas estruturas químicas.

Tabela 1.1: Simbologia e a nomenclatura dos aminoácidos.

Nome	Símbolo	Abreviação	Nomenclatura
Glicina ou Glicocola	Gly, Gli	G	Acido 2-aminoacético ou Acido 2-amino-etanóico
Alanina	Ala	A	Acido 2-aminopropiônico ou Acido 2-amino-propanóico
Leucina	Leu	L	Acido 2-aminoisocapróico ou Acido 2-amino-4-metil-pentanóico
Valina	Val	V	Acido 2-aminovalérico ou Acido 2-amino-3-metil-butanóico
Isoleucina	Ile	I	Acido 2-amino-3-metil-n-valérico ou ácido 2-amino-3-metil-pentanóico
Prolina	Pro	P	Acido pirrolidino-2-carboxílico
Fenilalanina	Phe ou Fen	F	Acido 2-amino-3-fenil-propiônico ou Acido 2-amino-3-fenil-propanóico
Serina	Ser	S	Acido 2-amino-3-hidroxi-propiônico ou Acido 2-amino-3-hidroxi-propanóico
Treonina	Thr, The	T	Acido 2-amino-3-hidroxi-n-butírico
Cisteína	Cys, Cis	C	Acido 2-bis-(2-amino-propiônico)-3-dissulfeto ou Acido 3-tiol-2-amino-propanóico
Tirosina	Tyr, Tir	Y	Acido 2-amino-3-(p-hidroxifenil)propiônico ou paraidroxifenilalanina
Asparagina	Asn	N	Acido 2-aminossuccionâmico
Glutamina	Gln	Q	Acido 2-aminoglutarâmico
Aspartato ou Acido aspártico	Asp	D	Acido 2-aminossuccínico ou Acido 2-amino-butanodióico
Glutamato ou Acido glutâmico	Glu	E	Acido 2-aminoglutárico
Arginina	Arg	R	Acido 2-amino-4-guanidina-n-valérico
Lisina	Lys, Lis	K	Acido 2,6-diaminocapróico ou Acido 2, 6-diaminoexanóico
Histidina	His	H	Acido 2-amino-3-imidazolpropiônico
Triptofano	Trp, Tri	W	Acido 2-amino-3-indolpropiônico
Metionina	Met	M	Acido 2-amino-3-metiltio-n-butírico

Fonte: dados do trabalho

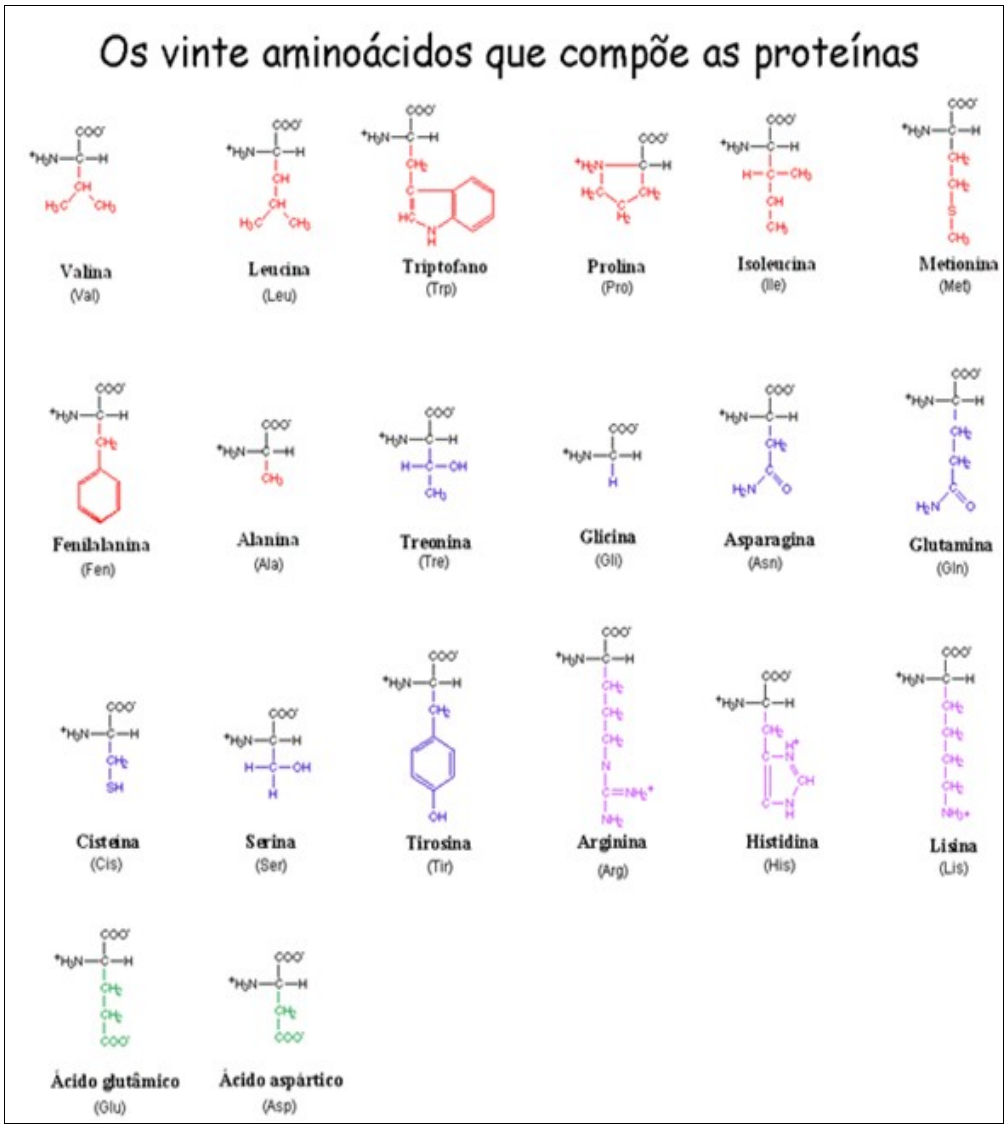

Figura 1.2: Estruturas dos 20 aminoácidos

Fonte: Notas de aula da disciplina de bioinformática DCC-UFLA ministrada em 2007

A molécula de uma proteína é um grande esqueleto peptídico (formado por ligações entre aminoácidos) com uma mistura de cadeias laterais de aminoácidos carregados, polares e apolares. O citoplasma é um ambiente aquoso, de maneira que as cadeias laterais carregadas e polares serão estabilizadas por interações como as moléculas de água do citoplasma. Entretanto, as cadeias laterais hidrofóbicas não interagem de maneira estável com a água, elas são mais estáveis quando estão agrupadas, afastados da água.

Kreuzer & Massey (2002) explica que parece que a regra básica que fundamenta a estrutura de proteínas é reunir as cadeias laterais dos aminoácidos hidrofóbicos, juntas, no interior da proteína, criando um ambiente hidrofóbico livre de água. Cadeias laterais hidrofílicas, ao contrario, são estáveis quando expostas ao citoplasma na superfície da molécula protéica. Isso não significa que você nunca encontrará um aminoácido hidrofóbico na superfície, mas em geral a regra é verdadeira. É dito, portanto, que uma proteína contém um núcleo hidrofóbico. A estrutura tridimensional de cada proteína individual pode ser pensada como uma solução para o problema de criar um núcleo hidrofóbico estável, dada a estrutura primária da proteína. Contudo, toda a estrutura protéica deve solucionar um problema comum.

Há um problema importante no dobramento de uma proteína para a criação de um núcleo hidrofóbico, o esqueleto. O esqueleto peptídico é repleto de ligações NH e CO, e ambas as ligações são altamente polares. Na superfície de uma proteína, essas ligações parcialmente carregadas podem ser prontamente neutralizadas por pontes de hidrogênio com água. Entretanto, para uma estrutura protéica ser estável, as cargas parciais do esqueleto peptídico dentro do núcleo da proteína, onde não há água, também devem ser neutralizadas. A solução para este problema é um fator crucial na determinação da estrutura protéica (KREUZER & MASSEY,2002).

A principal solução para o problema enfrentado pelo esqueleto peptídico dentro do interior hidrofóbico é a neutralização por suas próprias cargas parciais. Os grupos NH podem formar pontes de hidrogênio com os grupos CO, neutralizando ambos. Como cada aminoácido contribui com um grupo NH e um grupo CO para o esqueleto, esta solução é muito conveniente. Entretanto devido às restrições geométricas, os grupos NH e CO de um mesmo aminoácido não estão em posição para formar uma ponte de hidrogênio um com o outro. Em vez disso o esqueleto peptídico de ser cuidadosamente arranjado, de modo que os grupos NH e CO ao longo do esqueleto estejam em posição para formar pontes de hidrogênio com grupos complementares dispostos ao longo do esqueleto. Dois arranjos básicos funcionam bem e são os componentes principais da estrutura protéica (KREUZER E MASSEY,2002).

O primeiro arranjo para a autoneutralização do esqueleto peptídico é a formação de uma espiral helicoidal, como se ela estivesse se enrolando ao redor de um eixo imaginário. Os grupos NH e CO ao longo do esqueleto formam pontes de hidrogênio com grupos complementares situados acima ou abaixo deles, esse arranjo é chamado alfa-hélice (α-hélice).

No segundo arranjo para a autoneutralização, segmentos do esqueleto peptídico ficam lado a lado, de modo que um grupo CO em um esqueleto pode formar uma ponte de hidrogênio com um grupo NH do esqueleto adjacente. As cadeias laterais dos aminoácidos aparecem alternadamente acima e abaixo do plano do esqueleto. Esse arranjo é chamado folha Beta (folha β), e os segmentos individuais do esqueleto envolvidos na folha são chamados de fitas β. As folhas β, em geral não são planas, mas pregueadas.

As proteínas possuem uma propriedade muito importante que é a sua estrutura tridimensional, que por sua vez pode se classificar em diferentes estágios organizacionais, estrutura primária, estrutura secundária e estrutura terciária.

Segundo Dill (1990) ao estudar a estrutura da proteína pretende-se descobrir quais as propriedades da proteína que levam a cadeia a adotar uma estrutura única e estável e também, investigar como a seqüência de aminoácidos, estrutura primária, de uma proteína está relacionada com essas propriedades.

Como foi dito anteriormente as proteínas apresentam três tipos organizacionais de estrutura que se diferem de acordo com sua disposição espacial. Abaixo descreve esses níveis estruturais. Estrutura primária: é a representação da proteína por uma simples fita de aminoácidos sem apresentar nenhuma organização tridimensional. É o nível estrutural mais simples e mais importante, pois dele deriva o arranjo espacial da molécula. É possível observar a estrutura gráfica da estrutura primária pela Figura 1.3.

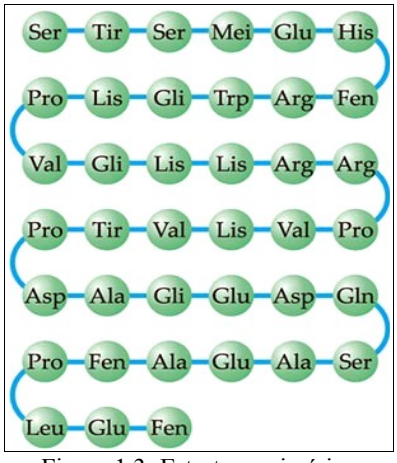

Figura 1.3: Estrutura primária

Fonte: Notas de aula da disciplina de bioinformática DCC-UFLA ministrada em 2007

Estrutura secundária: dentro de uma molécula de proteína, segmentos específicos da cadeia de aminoácidos podem assumir a conformação de uma α-hélice ou folha β. Certas seqüências de aminoácidos favorecem a formação de cada uma delas. Assim dentro do núcleo hidrofóbico de uma proteína, alguns segmentos do esqueleto podem ser encontrados em conformação de α-hélice, enquanto outros segmentos podem estar arranjados como folhas β. Algumas outras são formadas inteiramente por α-hélices e outras inteiramente por fitas β. Estas estruturas secundárias são freqüentemente conectadas umas às outras por meio de segmentos de aminoácidos na superfície da proteína, onde o esqueleto parcialmente carregado não precisa assumir uma estrutura secundária específica, já que ele é neutralizado pela água do meio celular. O sítio ativo das enzimas freqüentemente envolve estas alças desorganizadas de aminoácidos, provavelmente porque as alças são livres para mudar de conformação e se ligar a um substrato (BRANDEN & TOOZE, 1991).

A estrutura secundária de proteínas é dada pelo arranjo espacial de aminoácidos próximos entre si na seqüência primária da proteína, ela ocorre graças à possibilidade de rotação das ligações entre os carbonos dos aminoácidos e seus grupamentos amina e carboxila. O arranjo secundário de um polipeptídio pode ocorrer de forma regular, isso acontece quando os ângulos das ligações entre carbonos e seus ligantes são iguais e se repetem ao longo de um segmento da molécula.

As estruturas *α* são os elementos de estrutura secundária mais comum em uma cadeia polipeptídica dobrada, possivelmente porque são geradas por pontes de hidrogênio locais entre os grupos $C = O$ *e* $N - H$ próximos da seqüência. As *α-hélices* possuem número variado de aminoácidos, existem tanto *α-hélices* pequenas, com quatro ou cinco aminoácidos, quanto maiores, com mais de 40 aminoácidos. Em média seu comprimento é de 10 aminoácidos.

De acordo com Branden & Tooze (1991), As *folhas β* são o segundo maior grupo de estruturas formadas pela combinação de várias regiões da cadeia do polipeptídio. Elas possuem geralmente de cinco a dez aminoácidos. Existem duas formas em que as fitas de uma *folha β* podem interagir: a forma paralela e a antiparalela. A Figura 1.4 mostra alguns tipos de estruturas secundárias.

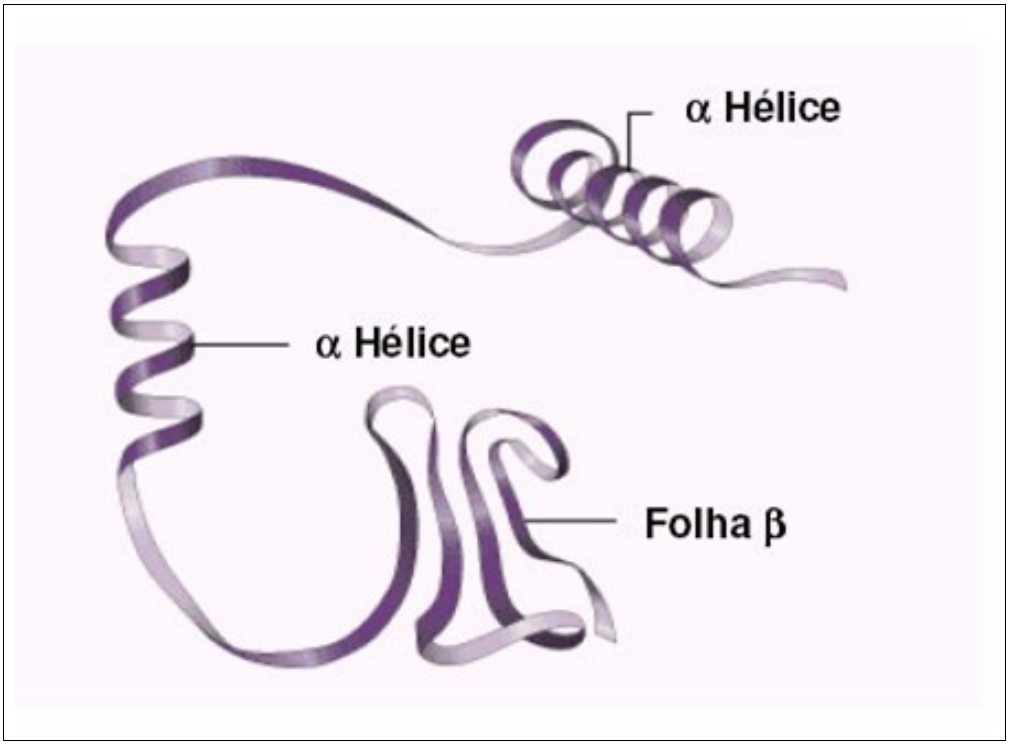

Figura 1.4: Estruturas secundárias

Fonte: Notas de aula da disciplina de bioinformática DCC-UFLA ministrada em 2007

Estrutura terciária: a estrutura terciária representa o arranjo tridimensional assumido por sua cadeia polipeptídica devido à composição das cadeias laterais dos aminoácidos. Este nível na hierarquia das proteínas refere-se também a como os elementos das estruturas secundária estarão dispostos no espaço tridimensional e como os aminoácidos interagem uns com os outros para formar pontes de hidrogênio. (BRANDEN & TOOZE, 1991)

Algumas combinações de estrutura secundária são feitas para formar uma estrutura tridimensional e compacta, chamada domínio. As proteínas pequenas podem ser constituídas de um único domínio, as proteínas maiores podem dobrar-se em vários domínios separados.

Os domínios parecem ser unidades fundamentais para a estrutura e função das proteínas. Os domínios são geralmente formados de seqüências contínuas de aminoácidos e, portanto, são traduzidos de regiões continuas de aminoácidos de mRNA. Em proteínas multifuncionais, não é incomum descobrir que a proteína se dobra em diferentes domínios e que cada domínio está associado a uma função. Em algumas situações, é possível separar os

domínios de uma proteína e, às vezes, podemos observar que os domínios separados retêm suas atividades individuais. A Figura 1.5 mostra a estrutura terciária.

Muitas proteínas funcionais são formadas por uma única cadeia de aminoácidos, porem muitas contêm mais do que uma cadeia polipeptídica. Estas cadeias podem ser diversas copias de uma mesma cadeia. Elas também podem ser a união de polipeptídios diferentes, a RNA polimerase de E. coli contém cinco cadeias diferentes, codificadas por cinco genes diferentes. A identidade, o número de cadeias polipeptídicas e a forma com que elas se unem na proteína final são chamados de estrutura quaternária da proteína. (KREUZER & MASSEY, 2002)

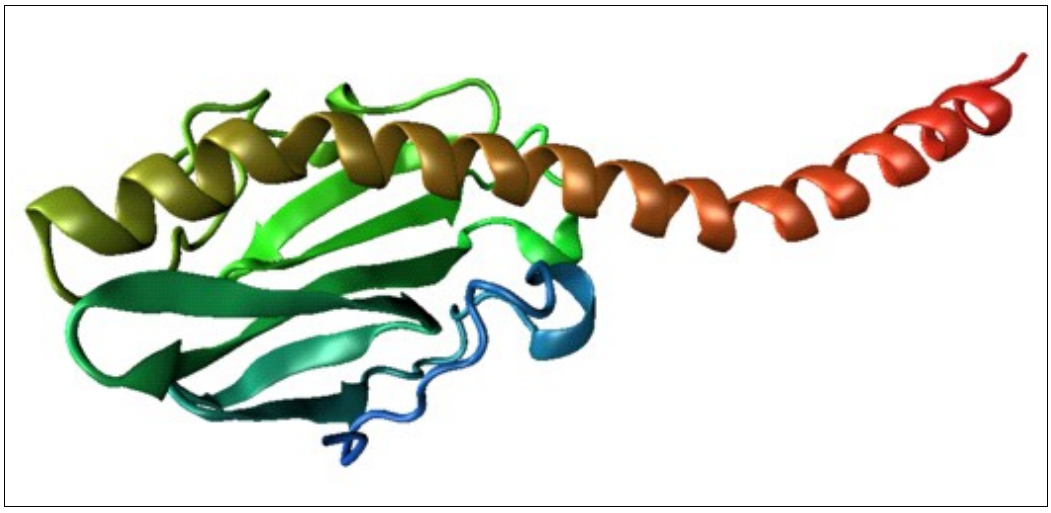

Figura 1.5: Estrutura terciária

Fonte: Notas de aula da disciplina de bioinformática DCC-UFLA ministrada em 2007

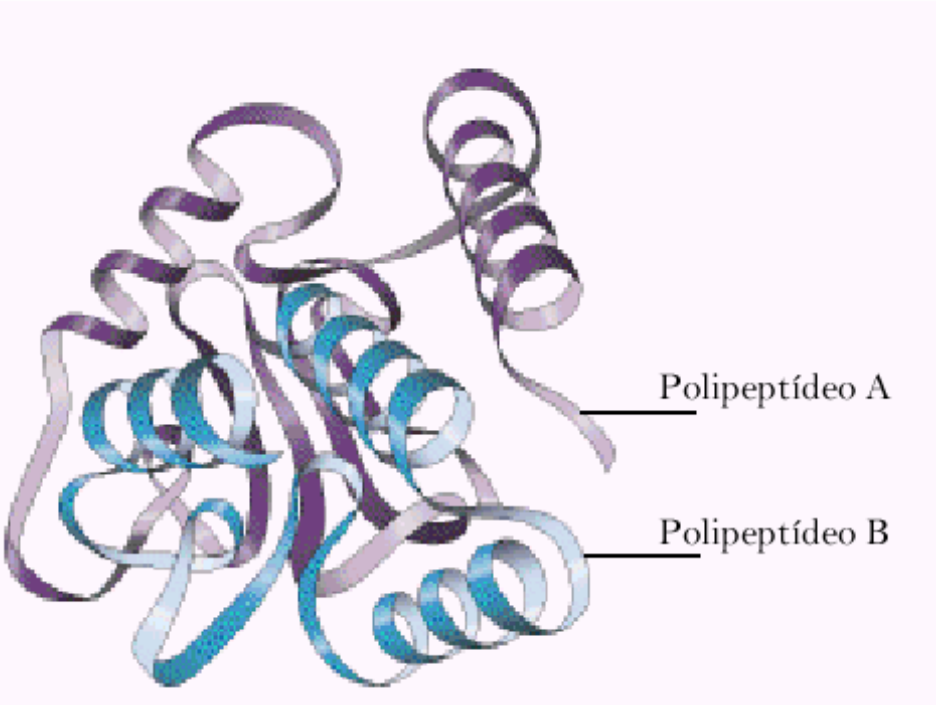

Figura 1.6: Estrutura quaternária

Fonte: Notas de aula da disciplina de bioinformática DCC-UFLA ministrada em 2007

1.2 Objetivos do Trabalho

Segundo Kreuzer & Massey (2002) prever a estrutura tridimensional de uma proteína a partir de aminoácidos é um dos maiores problemas ainda não resolvido da biologia molecular. Os biologistas estruturais vêm pesquisando a chave deste mistério nas estruturas catalogadas de proteínas, procurando por padrões de aminoácidos que se correlacionem a estruturas específicas. A busca de similaridades entre estruturas primárias é a forma mais comum de se inferir a estrutura tridimensional de uma nova proteína.

Ainda de acordo com Kreuzer & Massey (2002) a relação entre a seqüência de aminoácidos e a estrutura tridimensional tem sido chamada de a "segunda metade do código genético", pois é a estrutura tridimensional que conduz a função ao fenótipo Infelizmente, a relação entre a seqüência de aminoácido e a estrutura tridimensional não é trivial. Um dos caminhos é prever as estruturas secundárias de uma proteína para depois tentar prever a estrutura terciária.

Embasando-se nestes problemas este trabalho objetiva prever a estrutura secundária da proteína através de sua seqüência, obtidas de um banco de seqüências de proteínas. Para tal, serão utilizados recursos computacionais, neste caso redes neurais artificiais.

As redes neurais artificiais oferecem um bom suporte para prever as estruturas de proteínas, pois a sua propriedade de aprendizagem e de generalização garante bons resultados de previsões. O tipo de redes neurais utilizada é a rede multilayer perceptron com treinamento de taxa de aprendizado auto ajustável, com essa topologia de rede proposta espera-se conseguir obter melhores resultados, dos que já foram apresentados na literatura.

1.3 Organização do Texto

No capítulo dois é apresentada a noção de redes neurais artificiais, definições de topologias como características das redes neurais artificiais, aprendizado, generalização, abstração, estratégias de aprendizado, dados para treinamento, tamanho da rede neural artificial, pesos e parâmetros de aprendizado, mínimo local. E algumas informações importantes sobre redes neurais artificiais no ambiente de desenvolvimento MatLab.

Já no capítulo três é exposto como é o processo de predição de estruturas secundárias, com alguns algoritmos e resultados. No capítulo quatro tem-se a metodologia de trabalho, como foi realizado a obtenção dos dados, a classificação e a codificação, também será exposto a rede neural desenvolvida.

No capítulo cinco são apresentados os resultados esperados, e um comparativo com os resultados obtidos na literatura. E por fim no capítulo seis conclusão da pesquisa.

2 REDES NEURAIS ARTIFICIAIS

Para Braga, Carvalho & Ludermir (2007) redes neurais artificiais são sistemas paralelos distribuídos compostos por unidades de processamento simples, neurônios artificiais, que calculam determinadas funções matemáticas, normalmente não lineares. Tais unidades são dispostas em uma ou mais camadas e integradas por um grande número de conexões, geralmente unidirecionais. Na maioria dos modelos essas conexões estão associadas a pesos, os quais armazenam o conhecimento adquirido pelo modelo e servem para ponderar a entrada recebida por cada neurônio.

Barreto (2002) disse que redes neurais artificiais consistem em um modo de abordar a solução de problemas de inteligência artificial. Neste caso, em lugar de tentar programar um computador digital de modo a fazê-lo imitar um comportamento inteligente (saber jogar xadrez, compreender e manter um diálogo, traduzir línguas estrangeiras, resolver problemas de matemática, etc.) procura-se construir um computador que tenham circuitos, tais circuitos modelados como se fossem circuitos cerebrais e espera-se ver um comportamento inteligente emergindo, aprendendo novas tarefas, errando, fazendo generalizações e descobertas. Da mesma forma, estes circuitos neurais artificiais poderão se auto-organizar, quando apresentados a ambientes diversos, criando suas próprias representações internas e apresentar comportamentos imprevisíveis. E, melhor ainda, ou pior, ter um comportamento que nem sempre pode-se prever e compreender, tal como hoje não compreendemos mecanismos do nosso próprio cérebro.

Para Zuben (2003) redes neurais artificiais são sistemas de processamento de informação formados pela interconexão de unidades simples de processamento, denominadas neurônios artificiais. Os neurônios artificiais recebem essa denominação porque foram originados a partir de um modelo matemático de um neurônio natural. Além de sua natureza multidisciplinar, a computação inspirada em arquiteturas conexionistas não emprega os mesmos paradigmas predominantes na ciência da computação. A motivação que está por trás deste paradigma alternativo de processamento computacional é a possibilidade de elaborar soluções eficazes para problemas de difícil tratamento com base na computação convencional. O avanço verificado nos últimos anos junto à teoria de redes neurais artificiais tem levado invariavelmente ao desenvolvimento de ferramentas de engenharia mais eficazes e à utilização mais eficiente dos recursos computacionais hoje disponíveis, o que implica uma ampliação

sem precedentes na capacidade de manipular informação. O grande potencial das redes neurais artificiais só pode ser devidamente explorado com o emprego de procedimentos refinados de análise e síntese, requerendo assim um esforço adicional por parte dos usuários no sentido de aplicar os recursos de processamento disponíveis na medida certa e na situação apropriada..

Haykin (2001) relata que redes neurais artificiais representam uma tecnologia que tem raízes em muitas disciplinas: neurociência, matemática, estatística, física, ciência da computação e engenharia. Elas encontram aplicações em campos tão diversos, como modelagem, análises de séries temporais, reconhecimento de padrões, processamento de sinais, predições, em virtude de uma importante propriedade: a habilidade de aprender a partir de dados de entrada.

Redes neurais possuem certas características exclusivas de sistemas biológicos. Tais características entram em conflito com os tradicionais métodos computacionais. Sistema de computação baseado em redes neurais tem a capacidade de receber ao mesmo tempo várias entradas e distribuí-las de maneira organizada. Geralmente, as informações armazenadas por uma rede neural é compartilhada por todas as suas unidades de processamento. Característica que contrasta com os atuais esquemas de memória, onde a informação fica confinada em um determinado endereço. (BRAGA, CARVALHO & LUDERMIR, 2007)

Em um sistema de rede neural artificial, a informação pode parecer ter representação redundante, porém, o fato de que ela se encontre distribuída por todos os elementos da rede, significa que, mesmo que parte da rede seja destruída, a informação contida nesta parte ainda estará presente na rede, e poderá ser recuperada. Portanto, a redundância na representação de informações em uma rede neural, diferente de outros sistemas, transforma-se em uma vantagem, que torna o sistema tolerante a falhas. Atributos, tais como aprender através de exemplos, generalizações redundantes, e tolerância a falhas, proporcionam fortes incentivos para a escolha de redes neurais como uma escolha apropriada para aproximação para a modelagem de sistemas biológicos.(BRAGA, CARVALHO & LUDERMIR, 2007)

Lippman (1987) disse que o modelo de rede neural tem muitos neurônios conectados por pesos com capacidade de adaptação que podem ser arranjados em uma estrutura paralela. Por causa deste paralelismo, a falha de alguns neurônios não causam efeitos significantes para a performance de todo o sistema, o que é chamado de tolerância a falhas.

A principal força na estrutura de redes neurais reside em suas habilidades de adaptação e aprendizagem. A habilidade de adaptação e aprendizagem pelo ambiente significa que modelos de redes neurais podem lidar com dados imprecisos e situações não totalmente definidas. Uma rede treinada de maneira razoável tem a habilidade de generalizar quando é apresentada à entradas que não estão presentes em dados já conhecidos por ela.

A característica mais significante de redes neurais está em sua habilidade de aproximar qualquer função continua não linear de um grau de correção desejado. Esta habilidade das redes neurais as tem tornado útil para modelar sistemas não lineares na combinação de controladores não lineares.(LIPPMAN, 1987)

Redes Neurais podem ter várias entradas e várias saídas, eles são facilmente aplicáveis à sistemas com muitas variáveis.

2.1 Topologias das Redes Neurais Artificiais

De acordo com Másson (1990), a topologia de uma rede neural artificial pode ser expressa através de um grafo dirigido com pesos G = (V, A, W), onde V corresponde a um conjunto de vértices, A a um conjunto de arcos dirigidos e W a um conjunto de pesos para esses arcos. Cada vértice no grafo representa uma unidade de processamento.

As pesquisas em redes neurais artificiais levaram ao desenvolvimento dos mais diversos modelos cognitivos, cada qual com suas particularidades e adequados a um tipo de situação.

A estruturação de uma rede neural em camadas é uma importante característica topológica desses modelos. Em uma rede neural estruturada em camadas, o conjunto de vértices V pode ser particionado em vários subconjuntos disjuntos V = V(0) V(1) ... V(L) de modo que as unidades de processamento da camada l somente apresentem conexões com as unidades das camadas l+1 e l-1, onde l corresponde ao número de camadas da rede neural artificial.(MÁSSON, 1990)

Youngohc (1991) define uma rede *fully connected* como sendo aquela onde cada unidade de processamento da camada l estabelece conexão com todas as unidades de processamento da camada l+1.

Beale (1990) & Wasserman (1989) quanto ao número de camadas, as redes neurais podem ser dispostas em uma única camada, configuração mais simples de uma rede neural (single layer), ou em múltiplas camadas (multi layer).

Lippman (1987) ainda classifica as redes neurais artificiais em redes cíclicas, também chamadas de redes recorrentes, e redes acíclicas. A arquitetura de uma rede neural cíclica difere da acíclica por apresentar conexões entre as unidades de processamento pertencentes à mesma camada ou entre unidades de processamento de camadas diferentes cujas saídas passam a ser entradas na camada anterior. As redes recorrentes podem exibir propriedades muito similares à memória de curto termo dos seres humanos, onde o estado da saída da rede depende em parte da entrada anterior.

Másson (1990) disse que os arcos do grafo são chamados de conexões e representam as sinapses entre os neurônios artificiais. A cada conexão no grafo está associado um peso *wij(l)*, em analogia às sinapses de um modelo conexionista biológico, representando a força de ligação entre as unidades de processamento *vi(l)* e *vj(l*-1*)*, onde *i* e *j* correspondem à posição - respectivamente nas camadas *l* e *l*-1 , que essas unidades ocupam na rede .

Conexões com pesos positivos, chamadas excitatórias, indicam o reforço na ativação do neurônio *vi(l)*; sinapses com pesos negativos, chamadas inibitórias, indicam a inibição na ativação do neurônio *vi(l)*. Assim, os neurônios artificiais, distribuídos no espaço e ligados por conexões, trocam sinais inibitórios ou excitatórios, competindo ou cooperando entre si. O comportamento inteligente emerge, então, da ação simultânea dessa coletividade, sem a necessidade de elementos centralizadores (MÁSSON, 1990).

A interface da rede neural artificial é definida pelas unidades de entrada (V_I), unidades de saída (V_O) e pelas unidades ocultas (V_H) (MÁSSON,1990 & YOUNGOHC ,1991).

$$V = V_I\, V_O\, V_H$$

Freeman (1992) diz que o conjunto de unidades de entrada e de saída representa as unidades visíveis da rede, sendo dependentes da aplicação que se quer modelar. Em redes neurais estruturadas em camadas, a camada de entrada é tipicamente $V^{(0)}$ e a camada de saída, $V^{(L)}$.

Para Youngohc (1991) e Jones (1987) as unidades ocultas são utilizadas para modificar os dados de entrada, de modo a suportar qualquer função requerida para a entrada ou saída, impondo uma representação intermediária adicional dos dados de entrada para a saída desejada. Através das unidades ocultas, o modelo conexionista consegue representar abstrações que não poderiam ser diretamente realizadas a partir das unidades de entrada.

Para Beale (1990) o domínio do conhecimento de um problema é representado em um modelo conexionista através das unidades de processamento, que abstraem a estrutura e o comportamento dos neurônios biológicos.

Uma unidade de processamento $v_i^{(l)}$ possui entradas $x_1^{(l-1)}$, $x_2^{(l-1)}$, ..., $x_n^{(l-1)}$, que correspondem aos estados dos neurônios $v_j^{(l-1)}$ com os quais está conectada. A Figura 2.1 ilustra a estrutura de uma unidade de processamento.

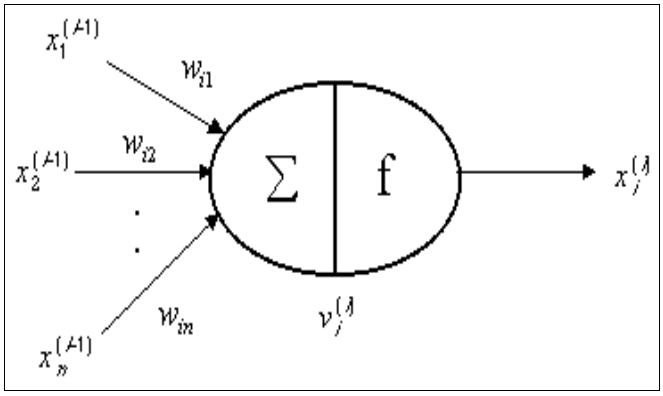

Figura 2.1: Unidade de Processamento

Fonte: dados do trabalho

De acordo com Másson (1990) a partir dessas entradas e do conjunto de pesos sinápticos $w_{ij}^{(l)}$, que refletem a força da unidade $v_j^{(l-1)}$ sobre a unidade $v_i^{(l)}$, é calculado o potencial *net* do neurônio $v_i^{(l)}$. Esse potencial no tempo *t* é determinado por uma *regra de propagação*, que geralmente equivale à soma linear da multiplicação das entradas pelos pesos , conforme equação

$$net_i^{(l)}{}_{(t)} = \sum_{j=1}^{n^{(l-1)}} w_{ij}^{(l)} x_j^{(l-1)}{}_{(t)} - \theta_i^{(l-1)}{}_{(t)}$$

onde $x_j^{(l-1)}$ é o estado da j-ésima unidade, $w_{ij}^{(l)}$ é a força sináptica entre a i-ésima unidade e a j-ésima unidade e, $_i^{(l)}$ é o limiar da i-ésima unidade, representando a força que as entradas das unidades conectadas à unidade $v_i^{(l)}$ precisam atingir para ativar esta unidade.

O potencial é modificado pela aplicação de uma *função de ativação g*, determinando o estado da unidade $v_i^{(l)}$ no instante *t+1*.

$$x_i^{(l)}{}_{(t+1)} = g\left(net_i^{(l)}{}_{(t)}\right)$$

De acordo com Másson (1990), a função de ativação corresponde a um limiar que restringe a propagação do impulso nervoso à transposição de um certo nível de atividade, mapeando o potencial da unidade de processamento $v_i^{(l)}$ para um intervalo pré-especificado de saída.

Dentre as possíveis funções de ativação pode-se citar a *linear*, a *rampa*, a *salto* e a *sigmóide*. A função *linear* (Figura 2.2 a) é obtida pela equação g () = $net_i^{(l)}{}_{(t)}$, onde é uma constante de proporcionalidade que regula a intensidade *do* $net_i^{(l)}{}_{(t)}$.

A função *rampa* (Figura 2.2 b) é limitada ao intervalo [-y,+y] definida por onde y representa o ponto de saturação da função.

$$g(net_i^{(l)}{}_{(t)}) = \begin{cases} +y, \text{ se } net_i^{(l)}{}_{(t)} \geq y \\ net_i^{(l)}{}_{(t)}, \text{ se } \left|net_i^{(l)}{}_{(t)}\right| < y \\ -y, \text{ se } net_i^{(l)}{}_{(t)} \leq y \end{cases}$$

A Figura 2.2 c) ilustra a função *salto* que admite valor +1 se o potencial da unidade de processamento $v_i^{(l)}$ for positivo e -1, caso contrário. A função *sigmóide* (Figura 2.2 d), também conhecida por função *logística*, é expressa matematicamente como

$$g(net_i^{(l)}{}_{(t)}) = \frac{1}{1+e^{-net_i^{(l)}{}_{(t)}}}$$

sendo uma função contínua, monotonicamente crescente e que gera valores graduais e não lineares no intervalo [0,1].

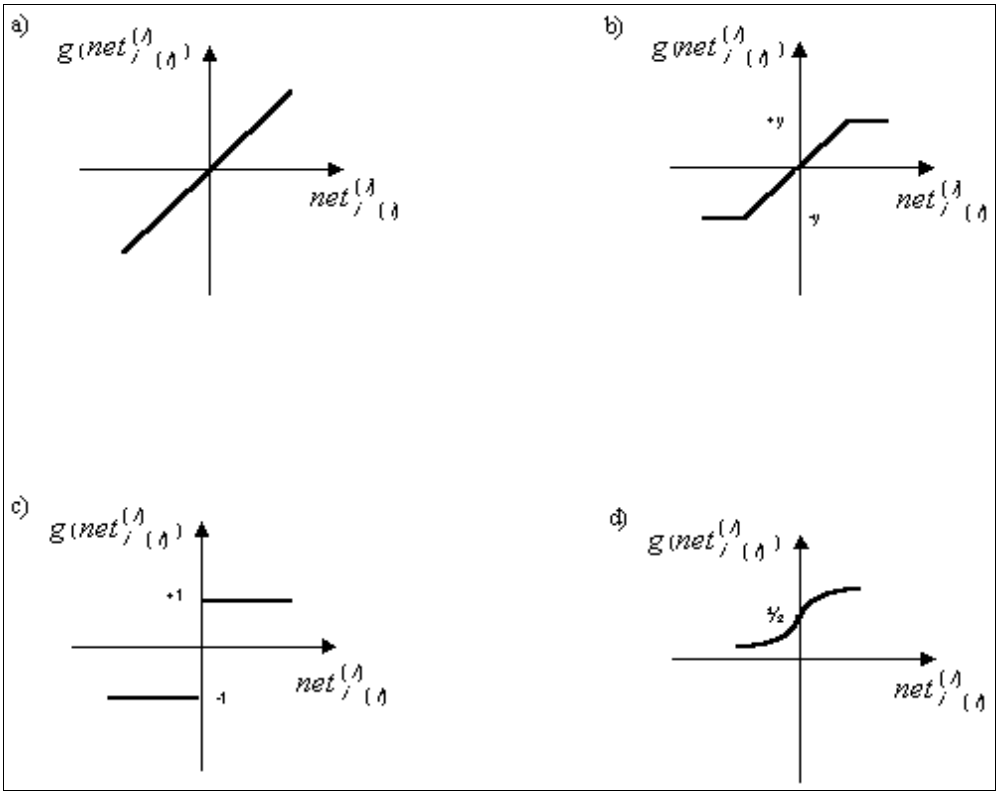

Figura 2.2: Funções de ativação

Fonte: dados do trabalho

Wasserman (1989) declara que outra função de ativação comumente utilizada é a *tangente hiperbólica* . Sua curva é similar à função sigmóide, mas simétrica na origem. Essa função é freqüentemente empregada por biólogos como um modelo matemático de ativação da célula nervosa.

Mesmo conhecendo completamente o comportamento de cada neurônio individual, a composição de várias unidades de processamento em uma estrutura de rede manifesta reações imprevisíveis. Desta forma, é a reunião do estado de ativação de todas as unidades de processamento que especifica o que está sendo representado na rede neural artificial em um determinado instante. É essa emergência de propriedades de conjunto que determina o interesse e a complexidade dos modelos conexionistas.

Por serem baseadas nas redes neurais biológicas, as redes neurais artificiais apresentam um surpreendente número de características observadas no processo cognitivo humano , como o aprendizado pela experiência, a generalização a partir de exemplos e a abstração de características essenciais de informações que contém fatos irrelevantes.(WASSERMAN, 1989)

Carbonell (1989) define o conceito de aprendizado como a habilidade de realizar tarefas novas que não podiam ser realizadas anteriormente, ou melhorar a realização de tarefas antigas, como resultado de mudanças produzidas pelo processo de aprendizado.

As redes neurais artificiais podem modificar seu comportamento em resposta aos estímulos produzidos pelo ambiente, regulando a força da conexão entre unidades de processamento adjacentes pela adaptação dos pesos sinápticos, reconhecendo as informações apresentadas às suas unidades visíveis (WASSERMAN, 1989).

Segundo Wasserman (1989), um modelo conexionista é sensível às variações que podem ocorrer em informações procedentes de suas unidades de entrada, reconhecendo ruído e distorção. A capacidade da rede em se adaptar às novas situações, gerando valores de saída consistentes com os esperados, é vital para a aplicabilidade do modelo em um ambiente do mundo real.

Refens (1993) disse que embora a maioria das pesquisas em redes neurais artificiais tenham concentrado seus esforços na redução dos tempos de aprendizagem, a característica mais importante de um modelo conexionista é a habilidade em generalizar sobre o domínio do problema.

De acordo com Lecun (1989) o bom desempenho da generalização depende, entre outros fatores, do número de parâmetros livres da rede neural artificial. É desejável diminuir o tamanho das conexões sem, entretanto, reduzir o tamanho da rede ao ponto onde não se possa computar a função desejada.

Alguns modelos de redes neurais artificiais são capazes de abstrair a essência do conjunto de dados a elas apresentados, permitindo, dessa forma, a classificação ou reconhecimento de padrões incompletos.

Dentre todas as características das redes neurais artificiais, nenhuma desperta tanto interesse quanto a sua habilidade em realizar o aprendizado (WASSERMAN, 1989).

Conforme Másson (1990), o aprendizado em um modelo de redes neurais artificiais é decorrente do treinamento da rede através da apresentação de padrões às suas unidades visíveis.

O objetivo do treinamento consiste em atribuir os pesos sinápticos com valores apropriados, de modo a produzir o conjunto de saídas desejadas ou ao menos consistentes com um intervalo de erro estabelecido. (FREEMAN,1992)

Desta forma, o processo de aprendizado subsiste na busca de um espaço de pesos pela aplicação de alguma regra que defina esta aprendizagem (MÁSSON, 1990).

Hebb (1949) define que, as regras de aprendizado podem ser consideradas variantes da Regra de Hebb. Na essência, Hebb propõe que a sinapse conectando dois neurônios seja reforçada sempre que ambos os neurônios estiverem ativos. Uma rede neural artificial que tenha a regra de Hebb como regra de aprendizado modifica os pesos sinápticos entre as conexões das unidades de processamento $v_i^{(l)}$ e $v_j^{(l-1)}$ proporcionalmente ao produto dos níveis de excitação desses neurônios, conforme equação,

$$\Delta w_{ij}^{(l)} = \eta . x_i^{(l)} . x_j^{(l-1)}$$

onde $\Delta w_{ij}^{(l)}$ corresponde a alteração no valor do peso $w_{ij}^{(l)}$ e é uma constante de proporcionalidade que reflete a evolução do processo de aprendizado pela busca no espaço de pesos.

Como adaptação à regra de Hebb, a regra Delta modifica os pesos de acordo com a variação entre a saída desejada e a observada no treinamento (MÁSSON, 1990).

A equação abaixo atualiza os pesos associados aos arcos da rede neural artificial pela aplicação da regra Delta

$$\Delta w_{ij}^{(l)} = \eta \ \delta \ x_i^{(l)}$$

Uma rede neural artificial deve ser ajustada para que a aplicação de um conjunto de entradas produza a saída desejada. Esse ajustamento, obtido pelo treinamento da rede, pode ser feito das seguintes formas (MÁSSON, 1990 & WASSERMAN, 1989).

- sem treinamento : os valores dos pesos sinápticos são estabelecidos explicitamente.
- treinamento supervisionado : a rede é treinada pela apresentação dos vetores de entrada e seus respectivos vetores de saída, chamados de pares de treinamento.

- treinamento não supervisionado : o treinamento consiste da apresentação apenas dos vetores de entrada, a partir dos quais são extraídas as características desse conjunto de padrões, agrupando-os em classes. O treinamento não supervisionado pode ser observado como um processo autônomo ou auto-organizável.

Por muitos anos não se teve um algoritmo eficiente para treinar redes neurais artificiais de múltiplas camadas. Desde que as redes de uma única camada se mostraram limitadas naquilo que poderiam representar e, portanto, no que poderiam aprender, o desenvolvimento de modelos cognitivos deixou de ser um campo atraente e poucas pesquisas foram realizadas na área.

O algoritmo *backpropagation*, proposto por Werbos, Parker e Rummelhart, fez ressurgir o interesse em redes neurais artificiais, sendo o algoritmo de aprendizado mais largamente utilizado. (MÁSSON,1990 e REFENES1993)

Conforme Beale (1990), o *backpropagation* pode ser visto como uma generalização do método Delta para redes neurais de múltiplas camadas. Ao se apresentar um determinado padrão de entrada a uma rede neural não treinada e o respectivo padrão de saída, uma saída aleatória é produzida. A partir da saída produzida pela rede é calculado um erro, representando a diferença entre o valor obtido e o desejado. O objetivo consiste, então, em reduzir continuamente o erro até um determinado valor aceitável. Isto é alcançado pelo ajuste dos pesos entre as conexões dos neurônios pela aplicação da regra Delta Generalizada, que calcula o erro para alguma unidade particular e propaga esse erro para a camada anterior. Cada unidade tem seus pesos ajustados de modo a minimizar o erro da rede.

A minimização do erro no algoritmo *backpropagation* é obtida pela execução do *gradiente decrescente* na superfície de erros do espaço de pesos, onde a altura para qualquer ponto no espaço de pesos corresponde à medida do erro. O ajuste dos pesos inicia nas unidades de saída, onde a medida do erro está disponível, e procede com a retropropagação desse erro entre as camadas, ajustando os pesos até que a camada das unidades de entrada tenha sido processada. Para as unidades de saída, como são conhecidos os valores desejados e obtidos, o ajuste dos pesos sinápticos é relativamente simples; para as unidades das camadas ocultas, o processo não é tão trivial. Intuitivamente, as unidades ocultas que apresentarem erros grandes devem ter suas conexões bastante alteradas, enquanto que a mudança nos pesos daquelas que tiverem suas saídas muito próximas das desejadas deverá ser pequena. Na realidade, os pesos

para um neurônio particular devem ser ajustados na proporção direta ao erro da unidade de processamento a qual está conectado. Essa é a razão pela qual a retropropagação dos erros através da rede permite o correto ajuste dos pesos sinápticos entre todas as camadas do modelo conexionista.

Assim, é possível identificar duas fases distintas no processo de aprendizagem do *backpropagation* : aquela onde as entradas se propagam entre as camadas da rede, da camada de entrada até a camada de saída, e aquela em que os erros são propagados na direção contrária ao fluxo de entrada.

Conforme Freeman (1992), não existe critério específico para seleção dos vetores de treinamento. É possível utilizar todos os dados disponíveis no treinamento do modelo conexionista, embora apenas um subconjunto desses dados talvez seja suficiente para que esse processo seja executado com sucesso. Os dados restantes podem ser usados para avaliar a capacidade de generalização do *backpropagation* no mapeamento de entradas nunca encontradas no treinamento para saídas consistentes.

Refens (1993) coloca que o número de unidades de processamento das camadas de entrada e saída é usualmente determinado pela aplicação. No caso das camadas ocultas, a relação não é tão transparente.

Rumelhart (1986) declara que o ideal é utilizar o menor número possível de unidades ocultas para que a generalização não fique prejudicada. Se o número de neurônios ocultos for muito grande, a rede acaba memorizando os padrões apresentados durante o treinamento. Contudo, se a arquitetura das camadas ocultas possuir unidades de processamento em número inferior ao necessário, o algoritmo *backpropagation* pode não conseguir ajustar os pesos sinápticos adequadamente, impedindo a convergência para uma solução.

Para Surkan (1990) a experiência ainda é a melhor indicação para a definição da topologia de um modelo conexionista.

Freeman (1992) sugere que os pesos das conexões entre as camadas de uma rede neural sejam inicializados com valores aleatórios e pequenos para que se evite a saturação da função de ativação e a conseqüente incapacidade de realizar a aprendizagem.

À medida que o treinamento evolui, os pesos sinápticos podem passar a assumir valores maiores, forçando a operação dos neurônios na região onde a derivada da função de ativação é muito pequena. Como o erro retropropagado é proporcional a esta derivada, o processo de treinamento tende a se estabilizar, levando a uma paralisação da rede sem que a solução tenha sido encontrada. Isto pode ser evitado pela aplicação de uma taxa de aprendizagem menor. Teoricamente, o algoritmo de aprendizado exige que a mudança nos pesos seja infinitesimal. Entretanto, a alteração dos pesos nessa proporção é impraticável, pois implicaria em tempo de treinamento infinito. Em vista disso, é recomendável que a taxa de aprendizado assuma valor maior no início do treinamento e, à medida em que se observe decréscimo no erro da rede, essa taxa também seja diminuída. Diminuindo progressivamente a taxa de atualização dos pesos, o *gradiente decrescente* está apto a alcançar uma solução melhor (BEALE, 1990 & RUMELHART,1986).

Beale (1990) e Freeman (1992) e Rumelhart (1986) e Wasserman (1989) dizem que outra maneira de aumentar a velocidade de convergência da rede neural artificial treinada pelo algoritmo *backpropagation* é a adoção de um método chamado *momentum* . O propósito desse método consiste em adicionar, quando do cálculo do valor da mudança do peso sináptico, uma fração proporcional à alteração anterior. Assim, a introdução desse termo na equação de adaptação dos pesos tende a aumentar a estabilidade do processo de aprendizado, favorecendo mudanças na mesma direção. A equação especifica o ajuste das conexões entre unidades de processamento pela aplicação do termo *momentum*. onde α representa o termo *momentum*, $0 < \alpha < 1$.

$$\Delta w_{ij}^{(l),\lambda}{}_{(t+1)} = \eta\ \delta_i^{(l),\lambda}\ x_j^{(l-1)} + \alpha\ (w_{ij}^{(l),\lambda}{}_{(t)} - w_{ij}^{(l),\lambda}{}_{(t-1)})$$

O *backpropagation* utiliza a heurística do *gradiente decrescente* para ajustar os pesos entre as sinapses, seguindo a curva da superfície dos erros em direção a um ponto mínimo. Superfícies de erros convexas, por apresentarem um único mínimo, permitem que este método atinja o mínimo global. Nas superfícies de erros não convexas e altamente convolutas, normalmente encontradas em problemas práticos, a solução alcançada pode não ser a ótima. Nestes casos, haverá que ser utilizado algum algoritmo de otimização global. (WASSERMAN, 1989)

Para Freeman (1992) assim que um mínimo é encontrado, seja global ou local, o aprendizado cessa . Se a rede alcançar um mínimo local (Figura 2.3), do seu ponto de vista

limitado, todas as direções em sua volta representam valores maiores que o alcançado e, conseqüentemente, a convergência para o mínimo global não é atingida. Nesse caso, a magnitude do erro da rede pode ser muito alta e, portanto, inaceitável.

Caso a rede neural encerre o aprendizado antes que uma solução satisfatória seja obtida, o redimensionamento do número de unidades ocultas ou da taxa de aprendizagem e do termo *momentum* podem ser suficientes para resolver o problema. Outra possibilidade para se tentar encontrar o mínimo global é realizar o treinamento a partir de um conjunto de pesos inicial diferente daquele utilizado anteriormente. (FREEMAN, 1992)

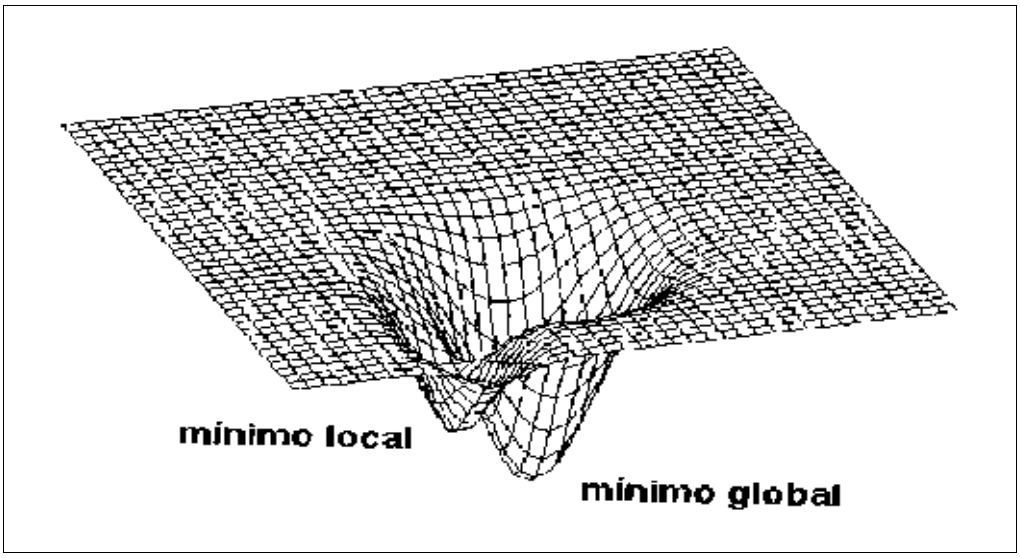

Figura 2.3: Mínimo Local

Fonte: Kovács (1996)

Através da Figura 2.3, que ilustra um corte em uma superfície de erros hipotética no espaço de pesos, é possível observar um ponto de mínimo local. Tanto à direita, quanto à esquerda, os valores são maiores que esse mínimo.

2.2 Redes Neurais Artificiais no MatLab

De acordo com Demuth & Beale & Hagan (2007) a escolha de usar o MatLab para fazer o treinamento da rede se deve ao fato que o MatLab apresenta boas funções para atacar o problema aqui abordado. Sua ToolBox para redes neurais artificiais contém varias opções de topologia de rede, e várias funções de treinamento diferentes e várias opções de mudanças de parâmetros, tendo assim uma vasta liberdade para obter bons resultados. Várias empresas e

instituições já utilizaram as funções de redes neurais artificiais do MatLab para suas pesquisas e produtos. Tendo isto em mente a opção pela utilização desta ferramenta é adequada.

Essa Tool Box para MatLab, apresentam vários tipos de modelos de neurônios, como o neurônio simples, como mostrado na Figura 2.4. Nela temos o modelo de neurônio simples utilizado pelo MatLab, à direita temos o modelo sem o baias como entrada tem-se um escalar *p*, ele é transmitido por uma conexão onde é multiplicado por um peso *w* para formar o produto *wp*, este produto *wp* é o único argumento para a função *f* que produz o escalar de saída *a*. E à esquerda temos o modelo com o baias onde o baias é o escalar *b*, nesse modelo temos a junção da soma do produto de *wp* com o baias, assim a saída a é a representação dessa soma. Para os dois modelos o argumento *f* representa a função de ativação, onde a função recebe o escalar *n* como parâmetro. Esse escalar é a representação da produto *wp* no caso do modelo sem baias ou do produto *wp* mais a soma do baias *b*, no caso do modelo com baias. (DEMUTH & BEALE & HAGAN, 2007)

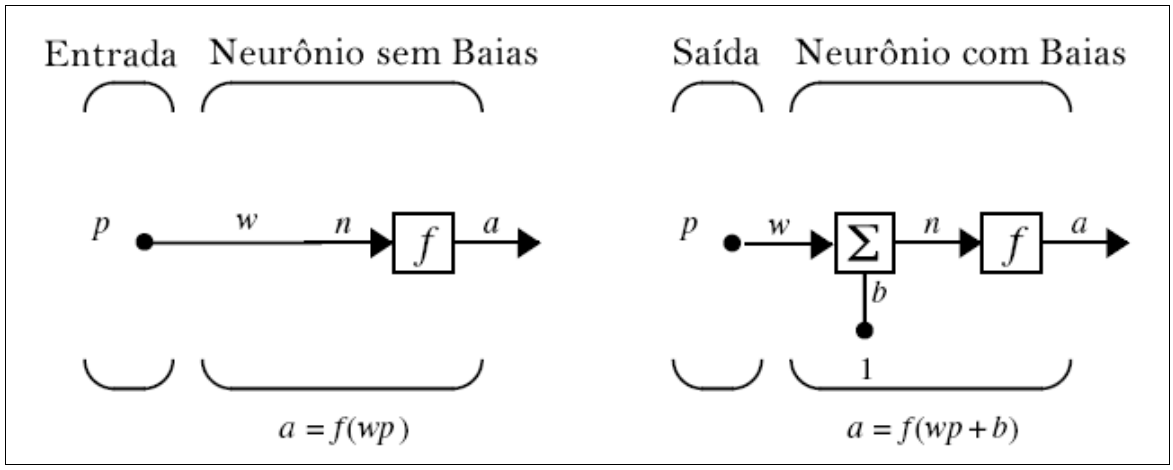

Figura 2.4: Modelo de neurônio simples

Fonte: Demuth & Beale & Hagan (2007)

Outro modelo de neurônio é o neurônio com vetor de entrada, mostrado na Figura 2.5 nesse modelo tem-se um vetor como entrada do neurônio cada elemento do vetor está representado por uma variável p, p_1 , p_2, p_3, ..., p_r onde estas entrada são multiplicadas pelos pesos $w_{1,1}$, $w_{1,2}$, ... $w_{1,r}$ esta é única diferença estre os dois modelos. Este modelo com vetor de entradas é o mais utilizado.

Nessa ToolBox apresenta também, várias funções de ativação para o treinamento das redes neurais artificiais, como mostrado na Figura 2.6.

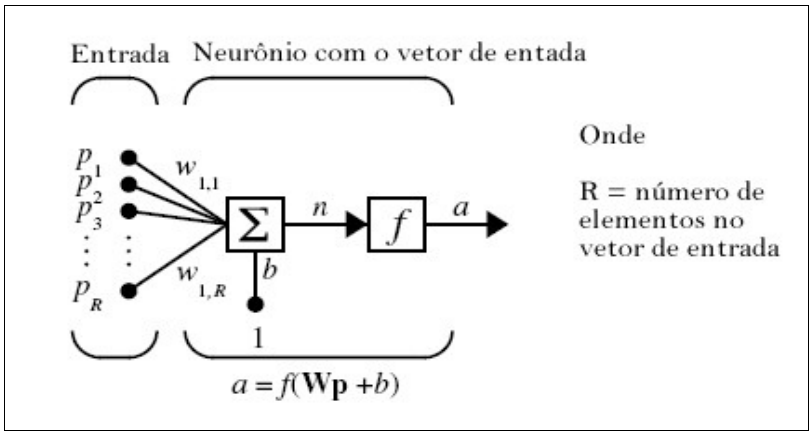

Figura 2.5: Neurônio com vetor de entrada

Fonte: Demuth & Beale & Hagan (2007)

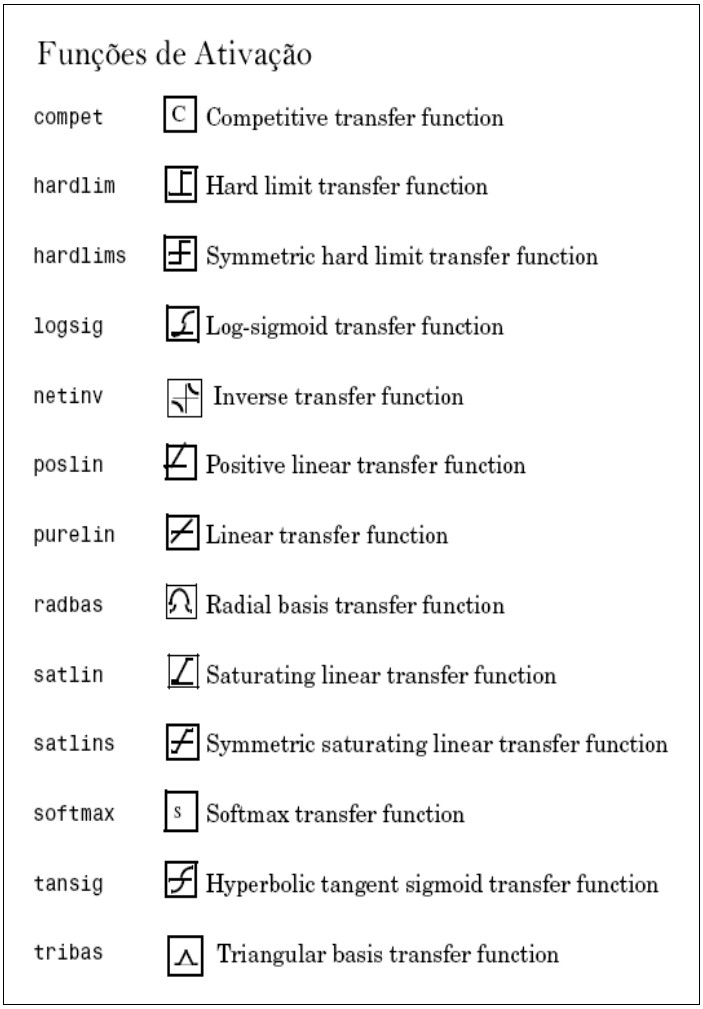

Figura 2.6: Funções de ativação

Fonte: Demuth & Beale & Hagan (2007)

Na figura à esquerda contém o nome da função de ativação no MatLab e a seguir uma pequena representação gráfica dela, e por fim o nome descritivo da função.

Para o Backpropagation o MatLab possui várias formas de treinamento. Treinamento pela função *Batch Training train,* nesse tipo de treinamento os pesos e o baias da rede são atualizados somente depois da entrada escolhida for aplicada na rede. Os gradientes calculados para cada exemplo treinado são adicionados juntos para determinar a mudança nos pesos e baias. Ainda para esse tipo de treinamento tem-se associado sete parâmetros, que são:

- *epochs*, parâmetro que determina a quantidade de épocas que vão executar durante o treinamento.
- *show*, parâmetro que determina o intervalo de épocas para atualizar o gráfico de performance.
- *goal*, parâmetro que determina o limiar para a performance da rede, caso com o treinamento ela atinja o valor desse limiar o treinamento é interrompido, antes mesmo se não tiver executados todas as épocas.
- *lr*, parâmetro que determina a taxa de aprendizado, essa taxa é multiplicada ao negativo do gradiente, para determinar as mudanças nos pesos e baias. Se essa taxa for muito grande, o algoritmo se torna instável, mas também se for muito pequeno o algoritmo demora muito tempo para convergir.
- *min_grad*, parâmetro que determina a magnitude mínima para o gradiente, a rede para o treinamento se ela obtém um gradiente mínimo escolhido por essa variável.
- *time*, parâmetro que determina o tempo em segundos para terminar a execução do treinamento.
- *mc*, parâmetro que determina a taxa de momentum, essa taxa é a responsável para que a rede não fique estacionada em um mínimo local.

Faster Tarining é outro tipo de treinamento que o MatLab possui. Este tipo de treinamento realiza técnicas de heurísticas ou otimização numérica para realizar o treinamento. O treinamento anteriormente mostrado pode ser uma boa opção, mas consome muito tempo para a execução dependendo do problema tratado. A ToolBox possui duas principais heurísticas que são a *traingda* e a *traingdx*, ou Variable Learning Rate, esses algoritmos

buscam em tempo de execução adaptar a taxa de aprendizagem. Um algoritmo por otimização é o Resiliente Backpropagation *trainrp.* Fletcher-Reeves Update *traincgf*, Polak-Ribiére Update *traincgp*, Powell-BealeRestarts *traincgb*, Scaled Conjugte Gradient *trainscg*, são treinamentos com otimização do gradiente. Já os BFGS Algorithm *trainbfg*, One Step Secant Algorithm *trainoss*, são algoritmos de treinamento com otimização pelo método de Newton. E por fim tem-se os métodos de Levemberg-Marquardt *trainlm,* e o Redeced Memnory Levenberg-Marquart *trainlm* os dois baseados no algoritmos de Levenberg-Marquardt. A Tabela 2.1 apresenta todos esses métodos de aprendizagem.

Tabela 2.1: Algoritmos para treinamentos de redes neurais artificiais

Acronym	Algorithm	
LM	`trainlm`	Levenberg-Marquardt
BFG	`trainbfg`	BFGS Quasi-Newton
RP	`trainrp`	Resilient Backpropagation
SCG	`trainscg`	Scaled Conjugate Gradient
CGB	`traincgb`	Conjugate Gradient with Powell/Beale Restarts
CGF	`traincgf`	Fletcher-Powell Conjugate Gradient
CGP	`traincgp`	Polak-Ribiére Conjugate Gradient
OSS	`trainoss`	One Step Secant
GDX	`traingdx`	Variable Learning Rate Backpropagation

Fonte: Demuth & Beale & Hagan (2007)

3 PREDIÇÃO DE ESTRUTURAS SECUNDÁRIAS DE PROTEÍNAS

Como foi dito no capítulo 1, a predição de estruturas secundárias de proteínas são importantes pois com elas podemos identificar quais são as funções desempenhadas pela proteína.

De acordo com Kaya (2008) estudos sobre predição de estruturas secundárias de proteínas por métodos computacionais e métodos estatísticos começaram em 1988 com Krigbaum e Kuntton, eles usaram algoritmos de regressão linear pra predizer, em seguida Chou-Fasman usou um método estatístico empírico baseado em freqüências de tipos de estruturas secundárias.

De acordo com Rost & Sander (1993) desde 1989 procura-se predizer as estruturas secundárias de proteínas por recursos computacionais. Mas até hoje o problema se mantém aberto, pois não se obteve um algoritmo que substitua totalmente os exames de laboratório.

Existem várias formas de se predizer a estrutura secundária a partir da estrutura primária. Uma das abordagens utilizadas é a predição por Redes Neurais Artificiais, o recurso computacional utilizado neste trabalho, mas também foram utilizados outras metodologias de predição como, Modelos de cadeia de Markov, Redes Bayesianas, Vector Machines. Este capítulo objetiva explicar superficialmente quais são esses métodos e quais são os resultados obtidos pelos mesmos.

Aydin & Altunbasak & Borodovsky (2006) disse que existem dois tipos de algoritmos para a predição. A primeira são algoritmos de predição para seqüências simples, que implica em não se ter o conhecimento de proteínas homólogas para realizar a predição. E o segundo tipo é quando se tem informações sobre proteínas homólogas.

As Redes Neurais Artificiais é o método computacional mais utilizado para predizer estruturas secundárias de proteínas, por sua capacidade de generalização se torna uma alternativa atraente para o combater o problema.

A predição por redes neurais artificias normalmente é modelada da seguinte maneira: as proteínas que já se tem o conhecimento de suas estruturas primárias e secundárias são utilizadas pela rede, a estrutura primária, como já foi dito, é a sua seqüência de aminoácidos e

será a entrada da rede, com se tem somente a informação dos aminoácidos que as compõe, alguma codificação dessa seqüência deve ser realizada. Afinal, redes neurais artificias só possuem dados numéricos como entrada. A estrutura secundária da proteína servirá como o vetor de valores esperados para a rede. Depois de definido a entrada e valores desejados, é preciso escolher a topologia, o algoritmo de treinamento e os ajustes dos parâmetros da rede, tais com número de épocas, função de ativação, número mínimo do gradiente, tempo máximo, e outros parâmetros que o algoritmo de treinamento tiver.

A codificação dos dados, a topologia da rede, o algoritmo de treinamento e os ajustes de parâmetros que são fatores necessários para se criar uma rede neural artificial para a predição de estruturas de proteínas deverá ficar a critério de escolha do pesquisador .

Qian & Sejnowski (1988) desenharam a primeira rede neural para realizar predições de estruturas secundárias de proteínas, conseguindo uma taxa de generalização de 64.5%. Em seguida Taylor & Orego (1989) conseguiram uma melhora taxa de acerto conseguindo chegar aos 65.5% de generalização. Outras pesquisas foram realizadas nos anos seguinte propondo outros algoritmos, mas nenhum trabalho se destacou, até que Rost & Sander (1993) introduziram na predição profiles alinhados com múltiplas seqüências alinhadas, o método foi chamado de PHD, e possuía performance bem melhor que as anteriores, pois utiliza alinhamento de profiles como entrada da rede, chegando a taxa 70%. Jones (1999) fez grades melhorias por ser pioneiro e usar a posição específica pontuando matrizes, esse método foi denominado PSSM com isso gerou-se o PSI-BLAST profile controlados, e logo em seguida pode-se criar o PSIPRED, a generalização desse método era um pouco superior a 70%.

McGuffin & Bryson & Jones (1999) disse que o PSIPRED server é um servidor de predição de estruturas de proteínas que está disponível em PSIPRED (2008) ele disponibiliza para usuários submissão uma seqüência de proteína os resultados da predição são enviados como uma mensagem de texto via e-mail, e graficamente via web. Esse servidor é constantemente atualizado, inserindo novos algoritmos e melhores resultados, hoje a taxa de exatidão do PSIPRED chega a 80% de exatidão.

Recentemente, novas técnicas de treinamentos e topologias de redes neurais artificiais tem sido freqüentemente usadas com intuito de obter melhores resultados para o problema da predição de estruturas secundárias de proteínas. Como por exemplo pode-se citar redes neurais

recorrentes, redes neurais holpfiel, redes neurais qprop, nprop, redes neurais com momentun. Com isso esses trabalhos conseguem cada vez mais, uma melhor eficiência nas predições.

Outro método para se prever estruturas secundárias de proteínas é por modelos de Markov. Um modelo escondido de Markov é uma máquina probabilísticas de estados finitos para modelos estocásticos de seqüências. O modelo de Markov é definido pelo conjunto de estados, emissão probabilística associado com cada estado conectado. Pode-se associar uma probabilidade com uma seqüência de acordo como o modelo de Markov que gera aquela seqüência..

Won & Hamelryck & Prugel-Bennett & Krogh (2005) disse que para usar um modelo de Markov para rotular uma seqüência é preciso associar uma etiqueta com cada estado, ou mais genericamente uma probabilidade de um marcador específico em virtude do estado. O rótulo atribuído a cada elemento na seqüência depende de qual estado que provavelmente tenha emitido o elemento. Esses modelos têm sido amplamente utilizados em bioinformática porque o conhecimento pode ser codificado para esses modelos. Alem disso, é permitindo que outras informações sejam aprendidas através de treinamentos das emissões e da transição das probabilidades dos dados.

Asai (1999) fez o primeiro modelo de cadeia de Markov para predição de estruturas secundárias de proteínas, com taxa de exatidão de 70%. Modelos de Markov com algoritmos genéticos foram desenvolvidos a fim obter melhores resultados.

O método de supoort vector machines proposto por Cortes & Vapnik (1995) é um método muito eficiente para reconhecimento de padrões , aprendizagem por support vector machines é a fronteira entre exemplos pertencentes a duas classes mapeando exemplos de entrada com um grande espaço dimensional, procurando um hiperplano de separação neste espaço. O hiperplano de separação é escolhido de forma a maximizar sua distância com relação aos exemplos de treinamento mais próximos. O hiperplano é chamado de separação hiperplana ótima.

Para Hua & Sun (2001) construir uma support vector machine para predizer estruturas secundárias de proteínas pode ser mais fácil que construir uma rede neural artificial. A estrutura apropriada da rede neural dependerá do nível do desenvolvedor, já no caso de support vector machines é necessário somente selecionar a função e regular um parâmetro para se poder

começar a treinar. Logo após é preciso determinar a janela de largura ótima para cada binário classificador. As taxas de exatidão para support vector machine, chegando a 77% obtida por Nguyen & Rajapakse (2007), demonstra a boa performance para a solução do problema.

4 MATERIAIS E MÉTODOS

Essa seção pretende esclarecer a classificação, e o caminho metodológico a ser percorrido para se alcançar os objetivos da pesquisa.

4.1 Tipo de Pesquisa

De acordo com Jung (2004), a pesquisa desenvolvida é aplicada, uma vez que se utiliza de conhecimentos e experiências adquiridos por estudiosos e profissionais da área de bioinformática e aplica técnicas já existentes na literatura.

Quanto ao objetivo, esta pesquisa é exploratória, visto que visa à descoberta de teorias e práticas que modificarão as existentes (JUNG, 2004).

Considerando-se os procedimentos a serem adotados, segundo Jung (2004) esta pesquisa é operacional, uma vez que aplica métodos científicos a problemas complexos para auxiliar no processo de tomada de decisões.

4.2 Definição do Problema

Pode-se observar durante a leitura dos capítulos anteriores que apesar de ser um problema muito visado na biologia computacional, a predição de estruturas secundárias de proteínas é um problema que precisa ser trabalhado. Os métodos aqui implementados visão dar um suporte à mais para os estudos no assunto. Para que algum dia o problema possa ser totalmente resolvido.

Os subcapítulos seguintes mostram detalhadamente os procedimentos, a obtenção dos dados a escolha da arquitetura da rede, a escolha da topologia e dos parâmetros, os resultados serão apresentados no capítulo seguinte.

4.3 Obtenção dos Dados

Para o treinamento da RNA foi utilizado o banco de dados público de proteínas Protein Data Bank, PDB, esse banco contém informações como o nome da proteína, sua seqüência (estrutura primária), possui informações sobre o tamanho a que estrutura secundária pertence essa seqüência. E outras muitas informações não relevantes para este trabalho. A Figura 4.1 mostra uma parte do PDB, pode-se observar que possui os campos referentes a quantidade de

aminoácidos(tamanho da seqüência), a seqüência e o campo DSSP que representa a estrutura secundária equivalente a cada aminoácido da seqüência. Para exemplificar, pode-se observar que a subseqüencia de aminoácidos FEMLRIDE a partir da quarta posição, representa uma estrutura secundária. Toda seqüência contínua de letras no campo DSSP representa uma única estrutura secundária. As estruturas alfa-helices são representadas pela letra H, as estruturas folha-beta são representas pela letra E e as estruturas Coils são representadas pela letra C, as demais letras são outros tipos de seqüências que não são tratadas neste trabalho.

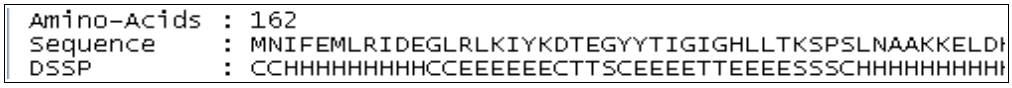

Figura 4.1: PDB - Protein Data Bank

Fonte: dados do trabalho

Para entrada de dados na rede, uma filtragem de dados foi realizada. Como a rede neural aceita somente como entrada seqüências de mesmo tamanho, uma seleção de seqüências de mesmo tamanho foi feita. Os dados foram separados por seqüências de tamanho dez, onde esses dez aminoácidos representam um tipo de estrutura secundária. A escolha do tamanho dez se deve ao fato que em média o tamanho das estruturas são de oito a doze aminoácidos, testes realizados obtiveram uma maior número de seqüências com tamanho dez.

Com a filtragem realizada pode-se observar que a quantidade de estruturas do tipo alfa-helice, folha-beta e coil são predominantes nas seqüências, foram catalogadas para a rede 90563 subseqüências, de tamanho dez, desses três tipos básicos, e 1047 subseqüências, de tamanho dez, de outros tipos.

Tabela 4.1: Número de subseqüências.

Estrutura	Número de subseqüências
Alfa-Helice	42564
Folha-Beta	28980
Coil	19019
Todas	90563

Fonte: dados do trabalho

A Tabela 4.1 mostra o número de subseqüências catalogadas depois da filtragem, pode-se observar que há um maior número de subseqüências alfa-helice, cerca de 47%, depois folhas-beta cerca de 32 %, e 21 % para Coil.

Como o tipo de estruturas secundárias mais encontradas são as alfa-helices as folhas-beta e as coil, a rede neural reconhecerá somente estes três tipos mais básicos de estruturas, o reconhecimento de outros tipos acarretaria em uma queda de performance da rede. A quantidade de subseqüências que não são destes três tipos é considerada insignificante, a rede simplesmente não os reconheceriam, reduzindo a performance.

A Figura 4.2 mostra a freqüência pelo tamanho para as seqüências. Observa-se que em média há uma maior freqüência de dados com o tamanho dez, para os três tipos básicos de estruturas secundárias, alfa-Helice, folha-Beta e Coil.

Outro fator importante para a rede neural é que as entradas além de serem do mesmo tamanho necessitam que os dados estejam em valores numéricos, e assim as seqüências filtradas passaram por uma codificação. Uma classificação por hidrofobicidade foi realizada, os aminoácidos recebem valores reais dependendo de seu grau hidrofóbico essa codificação também é denominada escala KD. A Tabela 4.2 expõe as codificações para cada um dos 20 aminoácidos. E também possui informações sobre que categoria de hidrofobicidade ele se encontra, que podem ser hidrofóbico, neutro ou hidrofílico. Pode-se observar que os valores começam em 0.05 e incrementam de acordo com os níveis de hidrofobicidade.

A codificação por hidrofobicidade foi escolhida pois como pode-se ver no capítulo 1, é de total importância para formação das estruturas o grau de hidrofobicidade dos aminoácidos. A formação de estruturas com a alfa-helice e a folha-beta dependerá se os aminoácidos que a constituem são ou não solúveis em meio aquoso.

Tabela 4.2: Valores reais atribuídos a cada aminoácido conforme escala de hidrofobicidade

Aminoácido	Escala KD	Valor Real	Categoria
I	+4,5	0,05	Hidrofóbico
V	+4,2	0,10	Hidrofóbico
L	+3,8	0,15	Hidrofóbico
F	+2,8	0,20	Hidrofóbico
C	+2,5	0,25	Hidrofóbico
M	+1,9	0,30	Hidrofóbico
A	+1,8	0,35	Hidrofóbico
G	-0,4	0,40	Neutro
T	-0,7	0,45	Neutro
S	-0,8	0,50	Neutro
W	-0,9	0,55	Neutro
Y	-1,3	0,60	Neutro
P	-1,6	0,65	Neutro
H	-3,2	0,70	Hidrofílico
Q	-3,5	0,75	Hidrofílico
N	-3,5	0,80	Hidrofílico
E	-3,5	0,85	Hidrofílico
D	-3,5	0,90	Hidrofílico
K	-3,9	0,95	Hidrofílico
R	-4,0	1,00	Hidrofílico

Fonte: dados do trabalho

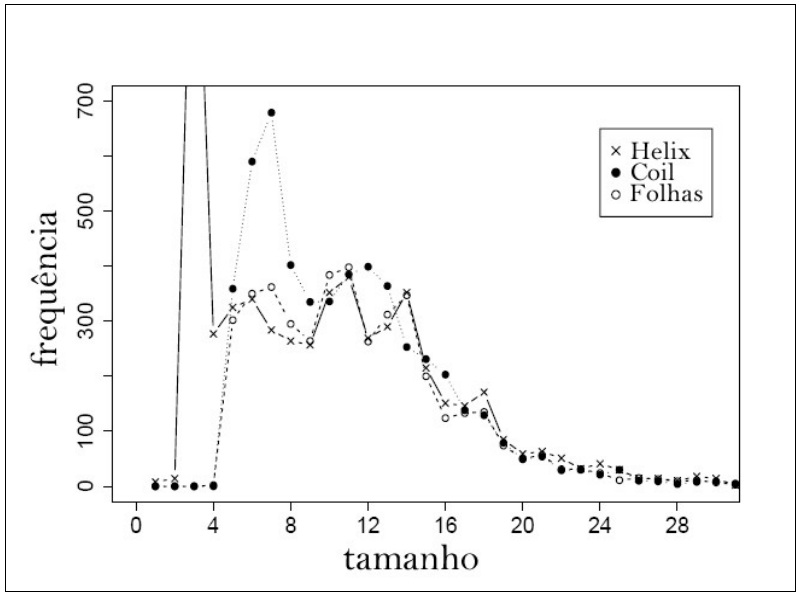

Figura 4.2: Freqüência por tamanho das estruturas secundárias

Fonte: Martin & Letellier & Marin & Taly & Brevern & Gibrat (2005)

4.4 A Rede Neural Artificial

Foram testados várias configurações de redes neurais artificiais, a fim de obter o melhor resultado, a configuração que obteve melhor resultado será aqui detalhada.

A Tabela 4.3 mostra um comparativo entre os resultados obtidos sobre as diferentes configurações de rede treinadas.

Tabela 4.3: Redes treinadas

Treinamento	Neurônios na camada intermediária	Função de ativação	Taxa total de generalização
train	10	tansig	70.6 %
trainbfg	10	tansig	72.3 %
traingrp	10	tansig	72.7 %
trainbfg	25	tansig	72.9 %
trainrp	25	tansig	73.6 %
trainbfg	30	tansig	74 %
trainrp	30	tansig	74.4 %
traingda	10	logsig	75.2 %
traingda	10	tansig	76.1 %
traingda	25	tansig	77.2 %
traingda	30	tansig	77.8 %
traingda	55	tansig	78.1 %

Fonte: Dados do trabalho

A melhor rede treinada, foi uma rede multi layer perceptron com treinamento backpropagation modificado com taxa de aprendizado adaptativa (treinamento Batch Training traingda do MatLab), feedforward, com taxa de momentum e funções de ativação tangente hiperbólica sigmoidal.

Em relação as camadas da rede, a primeira camada é a camada de entrada, esta é composta por dez neurônios cada neurônio representa um aminoácido e a junção destes dez aminoácidos (já codificados) é a representação de uma subseqüência que representam uma estrutura, cada vetor de entrada (os dez aminoácidos) pode representar uma estrutura diferente, podendo ser alfa-helice, folha-beta ou coil. Como foi escolhida uma topologia de rede multi

layer perceptron, MLP, a rede possui 55 neurônios na camada intermediária, essa configuração apresentou melhores resultados que configurações com menos neurônios nesta camada. E por fim tem-se três neurônios na camada de saída, cada um representando uma estrutura, entrando com uma subseqüência a rede diz se está é um dos três tipos de estruturas. Assim a rede neural artificial em relação a camadas tem a seguinte configuração: camada de entrada com 10 neurônios, camada intermediária com 55 neurônios e camada de saída com três neurônios, que pode ser visto na Figura 4.3.

O algoritmo de treinamento foi o traingda, ou seja algoritmo bacpropagation com taxa de aprendizado de modo adaptativo. O número de épocas executadas foram 6000, a taxa de momentum 0.5 e taxa de aprendizado 0.05. Esses foram os parâmetros da rede.

Dos dados obtidos, cerca de 70 % deles foram separados para o treinamento, e os outros 30% foram separados para validação da rede, quer dizer, 70% dos dados serão carregados como entrada e a rede neural artificial treinará com eles, os outros 30% servirão para a simulação, a verificação da performance da rede. Como pode-se observar a separação dos dados pela Tabela 4.1.

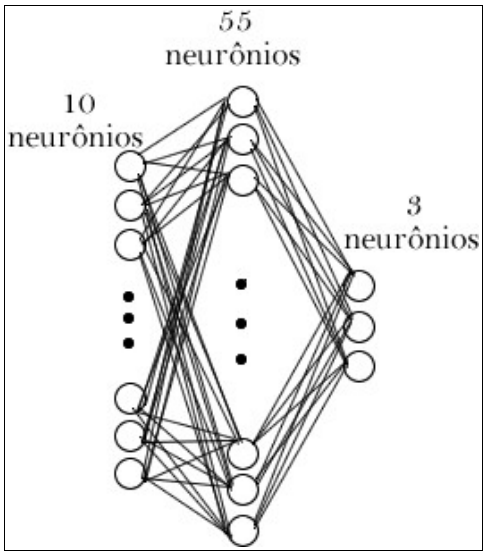

Figura 4.3: Arquitetura da rede.

Fonte: Dados do trabalho

Tabela 4.4: Dados para treinamento e dados para validação

Estrutura	Para 70% dos dados	Para 30% dos dados
Alfa-Helice	29794	12770
Folha-Beta	20286	8694
Coil	13313	5706
Total	63393	27170

Fonte: Dados do trabalho

4.5 Ambiente de Desenvolvimento

O trabalho foi realizado em um computador core 2 duo 1.86 Ghz, 1Gb de memória RAM, com sistema operacional microsoft windows XP service pack 3, o dados foram filtrados e codificados por um programa feito na linguagem Java, e a rede foi feita, treinada e simulada pelo MatLab com a toolbox de redes neurais artificias.

5 RESULTADOS E DISCUSSÃO

5.1 Os Resultados

O problema consiste na predição de estruturas secundárias de proteínas. Isso é, prever qual será a configuração de estruturas secundárias de uma dada proteína, através de sua estrutura primária, ou seja, através de sua seqüência de aminoácidos.

Para isto como já foi dito no capítulo 3, existem vários métodos de se prever essas estruturas secundárias de proteínas, dentre elas se destacam a predição por redes neurais artificiais, a predição utilizando métodos estatísticos, utilizando support vector machine, entre outros. Bons resultados já foram obtidos utilizando esses métodos, mas esse trabalho focalizou a obtenção das estruturas secundárias utilizando redes neurais artificiais.

Primeiramente ocorreu um tratamento dos dados, foi necessário realizar filtragens e codificações das seqüências de aminoácidos obtidos no banco de dados de proteínas, para que eles ficassem no formato permitido da rede neural artificial. Logo em seguida foi preciso identificar qual a topologia, a arquitetura e os parâmetros da rede. Com isso foi possível treinar a rede e depois de treinada, realizar simulações para obter os resultados.

Com o treinamento a rede obteve um erro de 0.106599. E com esse resultado obteve uma taxa de acertos totais de 78.1%, sendo que para Alfa-Helices a taxa foi de 89%, para folha-Beta a taxa foi de 77 % e de Coil a taxa foi de 68.3 %, para os 30% dos dados reservados à validação. Como pode ser visto na Tabela 4.4.

Tabela 5.1: Performance da rede.

Estrutura	Performance (%)
Alfa-Helice	89
Folha-Beta	77
Coil	68.3
Média	78.1

Fonte: Dados do trabalho

O resultado para Alfa-Helice foi o melhor resultado pois essas estruturas são as que apresentam maior volume de subseqüências. E a baixa performance para as coils se deve ao fato de uma quantidade reduzidas dessa estrutura nas subseqüências.

5.2 Comparativo dos Resultados

Como a taxa de generalização da rede foi de 78.1% ela obteve uma performance relativamente menor que aos que se encontram na literatura

A Tabela 5.1 mostras alguns dos melhores resultados obtidos na literatura. Os resultados podem ser obtido em: para o ID 1 Aydin & Altunbasak & Borodovsky (2006), para ID 2 Cuff & Clamp & Siddiqui & Finlay & Barton (1998), para o ID 3 Sen & Jernigan & Garnier & Kloczkowski (2005) , para o ID 4 Bondugula & Duzlevski & Xu (2005), para o ID 5 Hua & Sun (2001), para ID 6 Jones(2002), para o ID 7 Nguyen & Rajapakse (2007), para o ID 8 método implementado neste trabalho, para o ID 9 Pollastri & McLysaght (2004), para o ID 10 Peresen & Lundegaard & Nielsen (2000), para o ID 11 Zhou (2006), para o ID 12 Wood (2005), para o ID 13 Lin & Chang & Wu & Sung & Hsu (2005)

Tabela 5.2: Resultados para predições

ID	método	ano	performance (%)
1	cadeias de markov	2006	70.3
2	estatístico	1998	72.9
3	rede neural	2005	73.5
4	logica fuzzy	2005	75.75
5	support vector machine	2001	76.2
6	estatístico	2002	76.5
7	support vector machine	2007	77
8	rede neural	2008	78.1
9	rede neural	2004	79
10	rede neural	2000	80
11	rede neural	2006	80
12	estatístico	2005	80.7-81.7
13	rede neural	2005	81.8

Como o resultado deste trabalho, ID 8, foi uma taxa de generalização de 78,1%, um comparativo com resultados melhores do que o proposto aqui será detalhado. Para o algoritmo

9 Pollastri & McLysaght (2004) utilizou uma rede neural bidirecional recorrente com pequenas podas, assim obtendo um taxa de generalização de 79%. Para o algoritmo 10 Peresen & Lundegaard & Nielsen (2000) utilizou Posição específica de pontos em matrizes como entrada da rede, enquanto a saída com três estados consecutivos, a predição ocorreu com treinamnto por cross validation e foi testado em 1032 proteínas seqüênciadas, conseguindo assim uma taxa de generalização de 80%.Para o algoritmo 11 Zhou (2006) usou uma rede com larga escala de treinamento, em um cluster de alto desempenho com 22 processadores, a rede foi implementada com treinamento com cross validation e obterve uma taxa de generalização de 80%. Para para o algoritmo 12 Wood (2005) utilizou os resultados de um rede neural, feita por ele, que possuia uma um taxa de generalização de 79% com cross-validation, e aplicou métodos estatísticos e aplicou o método Ψ dihedral angles, assim obtendo 80.7 % a 81.7% de exatidão. E para o algoritmo 13 Lin & Chang & Wu & Sung & Hsu (2005) usou para cada aminoácido na proteína alvo, foi combinado os resultados do PROSP e PSIPRED usando uma função hibrida. Foram utilizadas duas bases de dados para o treinamento e a validação, o PDB e DSSPdataset, conseguindo assim uma taxa de generalização de 81.8%.

Mesmo não sendo a melhor solução para predição de estruturas secundárias de proteínas, o método aqui apresentado obteve um bom desempenho ao considerar que está entre os melhores resultados já obtidos.

Os resultados desse trabalho são bem melhores que os primeiros resultados, resultados que podem ser vistos no capítulo três, isso de deve ao fato que o número de proteínas que se tem conhecimento e que servem de base para realizar o treinamento é bem maior do que tinha naquela época, e métodos de treinamentos mais sofisticados, com heurísticas, também proporcionam melhores resultados.

6 CONCLUSÕES

Com esse trabalho pode-se concluir que a falta de informações sobre como foram realizados os processos de obtenção dos dados e tratamento , para os resultados obtidos na literatura não pode-se chegar a uma conclusão na diferença dos resultados. Para uma análise comparativa seria necessário ter todas essas informações, pois não se pode comprar processos onde os dados estão em formato diferente.

As limitações para este trabalho se tem com a falta de detalhes dos algoritmos disponíveis na literatura, fazendo com que o processo de reprodução ou de comparação aos resultados existentes se tornem difíceis.

A complexidade do problema o torna difícil de se tratar, ficando evidente pelo baixo nível dos resultados, onde a melhor predição encontrada leva a taxas de somente 81.8% de exatidão, somente com resultados melhores a predição poderia ser usada a fim de não mais precisar a utilização de métodos caros de laboratório para descobrir as estruturas de uma nova proteína descoberta ou catalogada.

E como trabalhos futuros poderá ser treinada redes separadas para os três tipos estruturas, uma rede treinada para alfa-helices, uma para folha-beta e outra para Coil, a fim de tentar melhorar a taxa de generalização. Assim construir preditores exclusivos para cada tipo de estrutura.

Também poderá realizar o treinamento da rede por outros algoritmos de treinamento com o algoritmo de treinamento Multi-Objetivo. A fim de tentar melhorar a performance dos resultados.

REFERÊNCIAS BIBLIOGRÁFICAS

ASAI, k.. Prediction of protein secondary structure by the hidden Markov model. , , v. , n. , p. , 1999.

AYDIN, Z. ALTUNBASAK, Y. BORODOVSKY, M.. Protein secondary prediction for a single-sequence using hidden semi-Markov models. , USA, v. , n. , p. , 2006.

BARRETO, J. M.; de F. M. AEVEDO; de LIMA, W. C. & ZANCHIN, C. I.. Neuralnetwork identification of resonance frequencies from noise.. In: . Vitoria, Brasil: , 2002. .

BEALE, R; JACKSON, T.. Neural Computing : an Introduction.. In: . : , 1990. .

BONDUGULA, R. DUZLEVSKI, O. XU, D.. Profiles and fuzzy k-nearest neighbor algorithm for protein secondary structure prediction. , , v. , n. , p. , 2005.

BRAGA, A. P., CARVALHO, A. P. L., LUDEMIR, T. B.. Redes Neurais Artificiais - Teoria e Aplicações. In: . : , 2007. .

BRANDEN, C. & TOOZE, J.. Introduction to Protein Structure. Garland Publishing. In: . : , 1991. .

CARBONELL, J. G.. Introduction : Paradigms for Machine Learning. In: . : , 1989. .

COPELAND, R.. Methods for Protein Analysis – A pratical guide to laboratory protocols. In: . : , 1993. .

CORTES, C. VAPNIK, V.. Support vector networks, machines learning. , , v. , n. , p. , 1995.

CUFF, J. A. CLAMP, M. E. SIDDIQUI, A. S. FINLAY, M. BARTON, G. J.. Protein secondary structure prediction based on position-specific scoring matrices. , , v. , n. , p. , 1998.

Demuth & Beale & Hagan (2007). . "Neural Network Toolbox 5 Use's guide". .

DILL, K A. Dominate Forces in protein Folding. In: . : Biochemistry, 1990. .

FONSECA, M. R. M. **Completamente química: química orgânica**. . São Paulo: Editora FTD, 2001. p.

FREEMAN, James A; SKAPURA, David M.. Neural Networks Algorithms, Applications and Programmming Techniques. In: . : , 1992. .

HAYKIN, S.. Redes Neurais. Principios e prática. In: . : , 2001. .

HEBB, D. O.. The Organization of Behavior.. In: . : , 1949. .

HUA, S. SUN, Z.. A Novel Method of Protein Secondary Structure Prediction with Segment Overlap Measure: Suppor Vector Machine Approach. , , v. , n. , p. , 2001.

JONES, D.. Protein secondary structure prediction based on position-specific scoring matrices. , , v. , n. , p. , 2002.

JONES, D. T.. Protein secondary structure prediction structure based on position-specific scoring matrices. , , v. , n. , p. , 1999.

JONES, William P.; HOSKINS, Josiah.. Back-Propagation : a Generalized Delta Learning Rule.. In: . : , 1987. .

JUNG, C. F. . Metodologia Para Pesquisa & Desenvolvimento. In: . : , 2002. .

KAYA, I. E.. Accurate Prediction of ProteinSecondary Structure By Non-Parametric Models. , India, v. , n. , p. , 2008.

kOVÁCS, Z. L.. **Redes Neurais Artificiais : Fundamentos e Aplicações**. . São Paulo: Editora Acadêmica , 1996. p.

KREUZER, H. MASSEY, A.. Engenharia Genética e Biotecnologia. In: . : , 2002. .

LE CUN, Y. . Generalisation and Network Design Strategies. In: . : , 1989. .

LEHNINGER, A. L. **Princípios de Bioquímica**. . São Paulo: Editora Sarvier, 1984. p.

LIN, H. N. CHANG, J. M. WU, K. P. SUNG, T. Y. HSU, W. L.. HYPROSP II-A knowledge-based hybrid method for protein secondary structure prediction based on local prediction confidence . , , v. , n. , p. , 2005.

LIPPMAN, R. P. An Introduction to Computing with Neural Nets.. In: . : IEEE ASSP Magazine, 1987. .

MARTIN, J. LETELLIER, G. MARIN, A. TALY, J. BREVERN, A. G. GIBRAT, J.. Protein secondary structure assignment revisited: a detailed analysis of different assignment methods. , , v. , n. , p. , 2005.

MÁSSON, Egill; WANG, Yih-Jeou. Introduction to Computation and Learning in Artificial Neural Networks. In: . North-Holand: European Journal of Operational Research, 1990. .

PSIPRED protein structure prediction server. Bioinformatics. Desenvolvido por: Bioinformatics, . . Disponível em: <http://bioinf.cs.ucl.ac.uk/psipred/psiform.html>. Acesso em: 25/10/2008.

MCGUFFIN, L.J. BRYSON, K. JONES, D. T.. The PSIPRED protein structure prediction server. , , v. , n. , p. 404-405, 1999.

NGUYEN, M. N. RAJAPAKSE, J. C.. Prediction of Protein Secondary Structure with two-stage multi-class SVMs. , , v. , n. , p. , 2007.

PERESEN, T. N. LUNDEGAARD, G. NIELSEN, M.. Prediction of protein secondary structure at 80% accuracy. , , v. , n. , p. , 2000.

PETSKO, G. and RINGE, D. . Proteins Structure and Function. New Science Press Ltd. In: . : , 2004. .

POLLASTRI, G. MCLYSAGHT, A.. Porter: a new, accurate server for protein secondary structure prediction. , , v. , n. , p. , 2004.

QIAN, N. SEJNOWSKI, T. J.. Predicting the secondary structure of globular proteins using neural network models. , , v. , n. , p. , 1988.

REFENES, A. N.; ALIPPI, C.. Histological Image Understanding by Error Backpropagation. In: . : Microprocessing and Microprogramming, 1993. .

ROST, B. SANDER, C.. Predictions fo protein secondary structure at better than 70% accuracy. , , v. , n. , p. , 1993.

RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. . Learning Internal Representations by Error Propagation.. In: . : Parallel Distributed Processing : exploration in the microstructure of cognition. Cambridge, 1986. .

SEN, T. Z. JERNIGAN, R. L. GARNIER, J. KLOCZKOWSKI, A.. GOR V server for protein secondary structure prediction . , , v. , n. , p. , 2005.

SURKAN, Alvin J; SINGLETON, Clay.. Neural Networks for Bond Rating Improved by Multiple Hidden Layers.. In: . San Diego: Proceedings of the IEEE International Joint Conference on Neural Networks, 1990. .

TAYLOR, W. R. OREGO, C. A. . Prediction of super-secondary structure in proteins. , London , v. , n. , p. , 1989.

WASSERMAN, Philip D.. Neural Computing : Theory and Practice. In: . New York: , 1989. .

WON, K. J. HAMELRYCK, T. PRUGEL-BENNETT, A. KROGH, A. . Evolving Hidden Markov Models for Protein Secondary Structure Prediction. , , v. , n. , p. , 2005.

WOOD, M. J.. **Protein secondary structure prediction with dihedral angles**. . : , 2005. p.

YOUNGOHC, Y.. Comparison of Discriminant Analysis vs Artificial Neural Networks. In: . : Journal of the Operational Research Society, 1991. .

ZHOU, Y.. Achieving 80% ten-fold cross-validated accuracy for secondary structure prediction by large-scale training. , , v. , n. , p. , 2006.

ZUBEN, F. J. V.. UMA CARICATURA FUNCIONAL DE REDES NEURAIS ARTIFICIAIS. In: . Campinas, Brasil: , . .