Universidade do Porto

Faculdade de Engenharia

FEUP

Redes Neuronais para Reconhecimento de Linguagem Gestual

Relatório Intercalar

Inteligência Artificial
3º ano do Mestrado Integrado em Engenharia Informática e Computação

Elementos do Grupo:
Diogo Joaquim Pinto – 201108016 - ei11120@fe.up.pt
Luís Brochado Pinto dos Reis – 201003074 - ei11009@fe.up.pt
Wilson da Silva Oliveira - 201109281 - ei11085@fe.up.pt

28 de Maio de 2014

Conteúdo

1 Objetivo **3**

2 Descrição **3**
2.1 Dataset 3
2.1.1 Parâmetros 3
2.1.2 Normalização dos dados 4
2.1.3 Processamento do Dataset 4
2.2 Arquitectura da Rede Neuronal 4
2.2.1 Backpropagation 5
2.3 Condição de paragem da aprendizagem 6

3 Estruturação da aplicação **6**

4 Medições **7**
4.1 Variação da velocidade de aprendizagem 7
4.2 Sucesso na aprendizagem 7
4.3 Resultados com a variação da quantidade de exemplos fornecidos 7

5 Conclusões **8**

6 Possíveis Melhorias **8**
6.1 Modificação do cálculo do erro 8
6.2 Cálculos através da GPU 8
6.3 Variância como parâmetro 8

7 Recursos **8**

1 Objetivo

Este projeto visa a criação de um programa que seja capaz de aprender e reconhecer a linguagem gestual Australiana (Auslan), utilizando dados com origem numa camera e num par de luvas com capacidade de efetuar medições de alta precisão, recorrendo às capacidades de aprendizagem não-simbólica das redes neuronais.

2 Descrição

2.1 Dataset

2.1.1 Parâmetros

Através da utilização da camera e das luvas de medição foram obtidos onze parâmetros diferentes para cada mão, perfazendo um total de vinte e dois parâmetros que serão processados pela aplicação.

- Posição X expressa em metros, em relação a uma origem definida ligeiramente abaixo do queixo;
- Posição Y expressa em metros, em relação a uma origem definida ligeiramente abaixo do queixo;
- Posição Z expressa em metros, em relação a uma origem definida ligeiramente abaixo do queixo;
- Rotação no eixo do X, medida num valor entre -0.5 e 0.5, sendo 0 a posição da palma da mão plana horizontalmente. Se o valor for positivo significa que a palma da mão está virada para cima na perspetiva do signatário. Para obter a medida em graus, multiplicar por 180;
- Rotação no eixo dos Y, medida num valor entre -1.0 e 1.0, sendo 0 a posição da palma para a frente na perspetiva do signatário;
- Curvatura do dedo polegar medida entre 0 e 1, sendo que 0 significa o dedo esticado e 1 o dedo totalmente dobrado;
- Curvatura do dedo indicador medida entre 0 e 1, sendo que 0 significa o dedo esticado e 1 o dedo totalmente dobrado;
- Curvatura do dedo médio medida entre 0 e 1, sendo que 0 significa o dedo esticado e 1 o dedo totalmente dobrado;
- Curvatura do dedo anelar medida entre 0 e 1, sendo que 0 significa o dedo esticado e 1 o dedo totalmente dobrado;
- Curvatura do dedo mindinho medida entre 0 e 1, sendo que 0 significa o dedo esticado e 1 o dedo totalmente dobrado.

É assinalado na fonte da base de dados que as medidas de dobra dos dedos não são totalmente exatas.

2.1.2 Normalização dos dados

O dataset necessitou de algumas normalizações, de modo a que todos os valores se encontrassem **entre 0 e 1**, nomeadamente em:

- X, Y, Z aplicou-se o módulo;
- Rotação no eixo do X incrementou-se 0.5 aos valores;
- Rotação no eixo do Y incrementou-se 1 e tirou-se a metade.

2.1.3 Processamento do Dataset

A base de dados contém um total de **95** palavras com **27** amostras para cada uma totalizando assim **2565** amostras. Cada amostra contém em média **60** medições únicas (correspondentes aos frames captados pelos dispositivos de medição) dos 22 parâmetros especificados anteriormente. Dado o elevado número de dados optamos por processar a média destes valores, numa tentativa de reduzir a carga sobre a rede neuronal, mas com o cuidado de garantir valores únicos para cada palavra diferente. Para tentar eliminar os frames menos essenciais para o gesto (que seriam os frames iniciais e finais), optamos por eliminar estes do cálculo da média final.

Para possibilitar o teste da aplicação nos nossos computadores "modestos"utilizamos uma versão reduzida da base de dados com apenas 8 palavras ao invés das **95** disponibilizadas. Não consideramos tal prejudicial visto que isto é uma *proof of concept*.

2.2 Arquitectura da Rede Neuronal

Existem 3 tipos de de rede neuronal, redes totalmente conectadas, redes de camada única e redes de múltiplas camada. Optamos pela implementação de **redes de camada múltipla**, visto que os dados não são linearmente separáveis; e permitiria a implementação de sub-redes, caso o dataset permitisse a associação entre parâmetros, o que levaria a um incremento na eficiência da rede neuronal; e é a qual, dado ser mais comummente utilizada, tem mais heurísticas e recursos desenvolvidos.

As camadas de entrada e de saída possuíam logo à partida valores fixos:

- a camada de entrada possui 22 nós, um por cada input;
- a camada de saída possui 8 nós (no caso da base de dados reduzida), um por cada palavra.

Após investigação sobre o tema com especialistas, não foi possível concluir qualquer associação entre os parâmetros relevantes.

Dado isto, foram testadas inúmeras arquiteturas, incluindo redes totalmente conexas com mais e menos nós na camada intermédia, bem como outras topologias: numa delas, como tentativa de agregação de informação, a camada intermédia recebia os dados dos nós da camada de entrada como se segue:

- posição (x, y, z) das duas mãos;
- dobra dos dedos polegar, indicador e médio da mão esquerda;

- dobra dos dedos polegar, indicador e médio da mão direita;
- dobra dos dedos anelar e mindinho da mão esquerda;
- dobra dos dedos anelar e mindinho da mão direita;
- rotações no espaço da mão esquerda;
- rotações no espaço da mão direita;
- dois nós com todos os parâmetros.

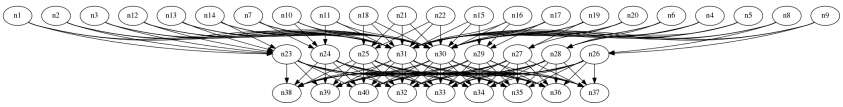

Figura 1: Primeira rede neuronal final

Após vários testes, acabamos com esta arquitetura como uma das finais: a outra é totalmente conexa e continha uma **camada intermédia única de 5 nós**.

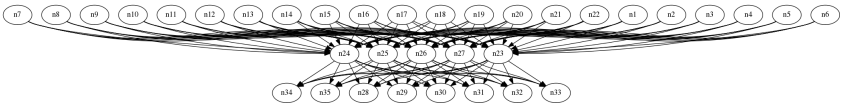

Figura 2: Segunda rede neuronal final

2.2.1 Backpropagation

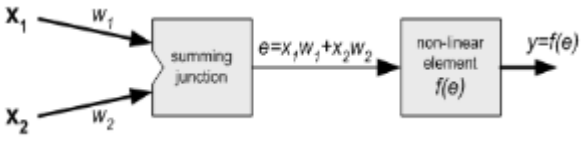

Figura 3: Exemplo de neurónio individual

O algoritmo de backpropagation é um método do tipo feedforward. Dado isto, inicialmente propaga-se os parâmetros recebidos na camada de entrada pelas ligações. Nas camadas intermédias e na camada de saída, om *impulsos* são somados e em seguida aplica-se uma função de transferência à medida que transmite os dados às camadas seguintes.

Chegando os *impulsos* à camada de saída, informação é retro-propagada para trás até à camada de entrada: após o cálculo dos erros na camada de saída, os erros são propagados para as camadas intermédias e à medida que o erro vai sendo propagado para trás, o peso das ligações entre camadas é alterado de modo a diminuir o erro. Este passo também é conhecido como "Treino"da rede.

$$s_i = \frac{1}{1+e^{\sum_{k=1}^{m} v_k * p_k}}$$

Figura 4: Transferência (sigmóide) com base na soma ponderada (pelos pesos) dos valores recebidos nas arestas

Atualizar o peso das ligações (é equivalente a minimizar o erro) é o modo de se reduzir para zero, ou aproximadamente zero, o erro na camada de saída, tornando a rede neuronal fidedigna.

$$\varpi_{it} = \varpi_{it-1} + \eta * \delta * \frac{\partial f(e)}{\partial e} * y$$

Figura 5: Cálculo do novo peso da aresta

Este processo é feito através da função de custo quadrático e da sua derivada, sendo que a primeira é a função a minimizar no algoritmo de retropropagação (apresentando $\boldsymbol{x}$ exemplos a um rede com $\boldsymbol{y}$ saídas).

2.3 Condição de paragem da aprendizagem

A aprendizagem é feita sobre a base de dados em ciclo, parando apenas quando uma iteração sobre todas as amostras de aprendizagem produz resultados corretos nos nós de saída. Definimos como resultados válidos na aprendizagem quando a diferença entre o valor mais alto na camada de saída difere de pelo menos **0.3** unidades do segundo mais alto. Já no teste, a diferença entre estes valores não tem qualquer restrição (de modo a potenciar a **generalização**).

3 Estruturação da aplicação

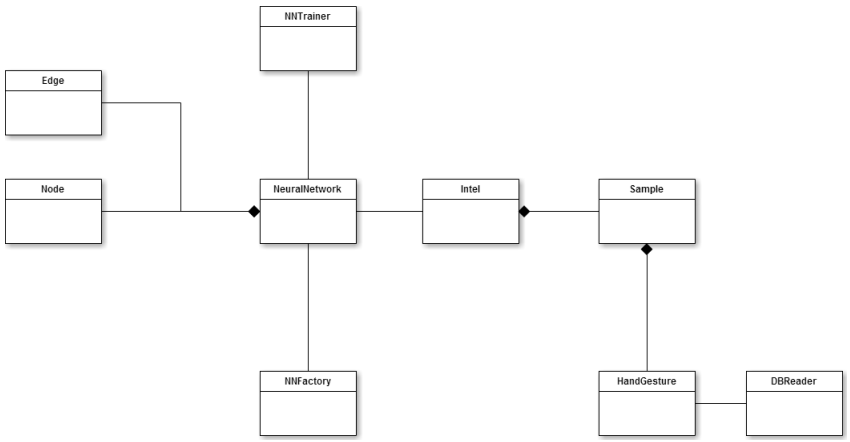

Figura 6: Diagrama de classes

4 Medições

4.1 Variação da velocidade de aprendizagem

Após vários testes, concluimos que o valor onde a aprendizagem era feita com menos flutuações, maximizando a rapidez da aprendizagem, era com o valor de **0.3**. Valores superiores implicavam uma aprendizagem deficiente.

4.2 Sucesso na aprendizagem

Foram testadas várias vezes as taxas de sucesso de aprendizagem, tendo sido efetuados testes com a base de conhecimento a 11%, 44% e 78% do total da base de dados. Concolui-se portanto que quanto maior for a base de dados, maior será o tempo que esta demora a ser aprendida e carregada, mas o sucesso de aprendizagem cresce em função do tamanho da base de dados. Em seguida apresentam-se os resultados dos testes realizados:

% da BD para aprendizagem	Tempo de treino(s)	Número de treinos	Sucesso (média)
11	87.9810	6819	26.646 %
44	271.000	5057	84.166 %
78	185.976	2113	95.833 %

4.3 Resultados com a variação da quantidade de exemplos fornecidos

Foi necessário verificar a validade de cada caso na seguinte equação:

$$L < S * E$$

L $\rightarrow$ nº de ligações
S $\rightarrow$ nº de saídas
E $\rightarrow$ nº de exemplos de aprendizagem

Sendo que

$$S * E = N$$

N $\rightarrow$ nº de equações

da BD para aprendizagem	Resultado	Verifica
11	$216 < 192$	Não
22	$216 < 384$	Sim
33	$216 < 576$	Sim
44	$216 < 768$	Sim
56	$216 < 960$	Sim
67	$216 < 1152$	Sim
78	$216 < 1344$	Sim
89	$216 < 1536$	Sim

É possível concluir que, para 11% da base de dados para aprendizagem, não existem exemplos suficientes para a rede neuronal **generalizar**, pelo que tende apenas a adaptar-se às equações lineares que modelam as soluções.

5 Conclusões

Após a realização deste projeto/investigação foi possível comprovar a influência da variação de parâmetros como a **velocidade de aprendizagem** e o **número de exemplos de aprendizagem**. Identificamos ainda a importância e o poder da capacidade de generalização e de tratamento de informação não simbólica das redes neuronais. Porém, esta mesma natureza leva a que não nos seja possível obter explicações para as conclusões que ela retira.

6 Possíveis Melhorias

6.1 Modificação do cálculo do erro

A inclusão no cálculo do erro de uma iteração de aprendizagem pela rede neuronal da derivada da função sigmóide levaria a uma aprendizagem acelerada nos valores intermédios e lenta perto dos valores limite (0 e 1), sendo que um exemplo já aprendido, no backpropagation não iria alterar em demasia os valores dos pesos, contribuindo para a estabilidade da rede.

6.2 Cálculos através da GPU

Uma hipótese de aumentar a rapidez de treino da base de dados seria implementar a execução dos cálculos através da GPU do computador.

6.3 Variância como parâmetro

Uma vez que os dados de input são a média das medições dos parâmetros das amostras, a adição do input das variâncias dos respectivos cálculos poderia ser uma mais valia na distinção entre amostras de movimentos dinâmicos e de estáticos. Seria preciso ter em conta, porém, que isto implicaria um aumento no número de arestas da rede, logo uma fase de aprendizagem mais prolongada.

7 Recursos

O software utilizado foi o IDE IntelliJ IDEA 13.1.1, para programação em Java.

Referências

[1] Eugénio Oliveira *Diapositivos de IART:* `http://paginas.fe.up.pt/~eol/IA/1314/APONTAMENTOS/7_RN.pdf` 2013-2014.

[2] Wikipedia, Redes Neuronais Modulares `http://en.wikipedia.org/wiki/Modular_neural_network`.

[3] Peter Norvig, Stuart Russell *Artificial Intelligence A Modern Approach* 2009: Prentice Hall.