Relatório do Primeiro Problema - EXA868 Inteligência Artificial Não-Simbólica B

Guilherme Lopes Wanderley [1] **e Manuella dos Reis Araújo Vieira** [1]

[1]DTEC – Universidade Estadual de Feira de Santana (UEFS)
Caixa Postal 44036-900 – Feira de Santana – BA – Brazil

Resumo. *Este relatório descreve o desenvolvimento de uma rede neural MLP (perceptron multicamadas) para classificação de comentários da plataforma de jogos digitais Steam nas categorias positivo e negativo, aplicando conceitos de mineração de opinião e análise de sentimentos.*

1. Introdução

Este trabalho objetiva descrever o desenvolvimento de uma rede neural MLP (perceptron multicamadas) com uma camada escondida, desenvolvida para a resolução de um problema de mineração de opinião e análise de sentimentos. A rede deve ser capaz de classificar comentários de avaliação de produtos nas categorias positiva ou negativa, com base na análise dos textos escritos pelos usuários.

Para este trabalho, foram escolhidos como base de análise comentários da plataforma de venda de jogos digitais Steam. O sistema de classificação da Steam permite que o usuário, além de expressar sua opinião textualmente, indique se o jogo é Recomendado ou Não Recomendado por ele. Sendo assim, a rede deve ser capaz de, após o treinamento, indicar se um determinado comentário é positivo (recomendado) ou negativo (não recomendado) a partir do texto.

2. Ferramentas Utilizadas

A rede neural foi desenvolvida na linguagem de programação Python, com o auxílio da biblioteca TensorFlow. O TensorFlow é um *framework* de aprendizado de máquina, podendo ser utilizado para criar e treinar redes neurais. Para geração dos gráficos, foi utilizada a biblioteca matlibplot.pyplot.

3. Conceitos Teóricos

Nesta seção, serão brevemente descritos conceitos teóricos aplicados ao projeto da rede neural.

3.1. Época

O conceito de épocas de treinamento é uma medida do número de vezes em que todos os dados de treinamento são usados uma vez para atualizar os valores dos pesos. A cada vez que uma rede neural é treinada para todos os exemplos de entrada, uma época é completa.

3.2. Paramêtros

Os parâmetros de um modelo de rede neural são as variáveis utilizadas para ajustar os dados da rede, que são modificadas ao longo do processo de treinamento, como os pesos.

3.3. Hiperparâmetros

Hiperparâmetros são variáveis de configuração, que contêm os dados que governam o próprio processo de treinamento, como a quantidade de camadas escondidas a serem utilizadas e quantos nós cada camada deve possuir.

4. Desenvolvimento

O desenvolvimento da rede neural passou pelas etapas de coleta de dados, pré-processamento dos mesmos - para que estes possam servir de entrada para a rede - e implementação efetiva da rede, estabelecendo configurações iniciais como função de ativação e quantidade de neurônios na camada escondida.

4.1. Coleta de Dados

Os dados utilizados para treinamento, validação e teste da rede neural foram retirados do repositório https://github.com/mulhod/steam-reviews no GitHub, no formato JSON. Os dados foram coletados para os seguintes jogos (entre parênteses, o número de comentários extraídos):

- Arma 3 (7,151)
- Counter Strike (6,040)
- Counter Strike: Global Offensive (7,073)
- Dota 2 (9,720)
- Football Manager 2015 (1,522)
- Garry's Mod (7,151)
- Grand Theft Auto V (13,349)
- Sid Meier's Civilization 5 (7,467)
- Team Fortress 2 (5,676)
- The Elder Scrolls V (7,165)
- Warframe (7,123)

Os dados encontrados no repositório incluem uma série de informações sobre a análise feita pelo usuário, como o número de pessoas que acharam a análise útil, o número de horas que o usuário passou jogando o jogo revisado, dentre outras. Visto que as únicas informações relevantes para a rede neural tratada neste relatório são a análise textual e a indicação de recomendação do jogo, outras informações foram eliminadas para processamento posterior dos dados. Os comentários de cada jogo foram analisados e processados separadamente.

4.2. Pré-processamento

4.2.1. Lavagem de Dados

Do texto, todas as palavras foram convertidas para letra minúscula e os símbolos foram retirados. Estes elementos são elementos de marcação para leitura de seres humanos, portanto são descartados neste trabalho. Deste modo, a máquina é capaz de detectar que duas palavras são iguais, mesmo que uma esteja adjacente a um símbolo, como vírgula ou ponto.

4.2.2. Divisão dos Dados

Em seguida, os textos de cada jogo foram separados em três conjuntos: treino, validação e teste. Os dados foram divididos em 90% para treino e 10% para teste, sendo que dos 90% de treino, 10% são utilizados para validação. Desta forma, o conjunto de treino possui 89% dos dados, o conjunto de validação possui 1% dos dados e o conjunto de teste possui 10% dos dados.

Os conjuntos de treino, validação e teste possuem equilíbrio nos dados, sendo metade dos dados positivos e a outra metade negativos. Por conta da diferença de proporção entre comentários positivos e negativos, os conjuntos foram divididos de acordo com o número de comentários negativos, pois estão em menor quantidade.

4.2.3. Representação dos Dados

Redes neurais recebem apenas números como entrada, no entanto, estamos analisando textos. Para converter de palavras para números, todas as palavras precisam receber um número. Para a palavra ser considerada relevante a ponto de receber um número, deve aparecer ao menos 5 vezes em todo o documento. A palavra correspondente ao número 0 é usada para palavras desconhecidas, que não foram aprendidas durante a etapa de treino da rede. Estes números são normalizados em um vetor, onde o índice do vetor corresponde ao número da palavra e os valores 1 ou 0 indicam presença ou ausência da palavra, respectivamente. Cada comentário é traduzido em um vetor e o conjunto de comentários se torna uma matriz. Essa representação é conhecida como *One Hot Vector*.

4.3. Configurações Iniciais da Rede

A arquitetura da rede é composta por três camadas, sendo uma de entrada, uma camada escondida e uma de saída.

A função de ativação *softmax* - dada pela equação da Figura 1 - foi escolhida em detrimento da função sigmóide, com o intuito de eliminar um parâmetro, já que a função *softmax* calcula a probabilidade de uma entrada pertencer à classe negativa ou positiva e a função sigmóide calcula um número, sendo necessário estabelecer um valor de corte para identificar um comentário como positivo ou negativo.

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}$$

Figura 1. Função softmax

A rede foi iniciada com alguns parâmetros inicias, que são variados em cada experimento. Os pesos iniciais variam no intervalo [-0,7:0,7]. A quantidade inicial de neurônios na camada escondida é de 256. A rede deve passar por 30 épocas de treinamento e cada passo do treinamento considera um batch de 100 comentários. A taxa de aprendizagem é de 0,1. Estes valores foram determinados com base em outros trabalhos divulgados sobre implementações de Redes Neurais. A dimensão do vocabulário, que

determina a quantidade de linhas da matriz de pesos, é de 1326, retirada dos comentários do jogo Football_Manager_2015.

4.4. Experimentos

Os experimentos consistem no treinamento e coleta dos resultados de redes neurais.

Os parâmetros foram variados na seguinte ordem: pesos iniciais (valores próximos), pesos inicias (valores distantes), número de neurônios, taxa de aprendizagem, tamanho do batch, momentum e otimizadores.

Para que os pesos inicias tenham valores distantes, variou-se a média e o desvio padrão da função de inicialização randômica. Isto aumenta cada vez mais o valor absoluto e a amplitude dos valores iniciados.

Para variação do número de neurônios, também foram variados os pesos iniciais, uma vez que estas medidas são interdependentes.

São realizados 20 treinos variando um único parâmetro para coleta de dados. Em todos os treinos é coletada a acurácia do modelo em relação ao conjunto de teste. A cada 3 treinos, a partir do primeiro, são coletados dados de erro e acurácia do modelo no conjunto de treino.

5. Resultados

Os resultados são exibidos e discutidos a partir de gráficos gerados pelos experimentos. Cada item na legenda do gráfico é composto por dois números. O primeiro número corresponde ao valor do parâmetro sendo variado e o segundo número corresponde à acurácia final daquela variação no conjunto de teste. Os eixos possuem identificação e variam de acordo com o experimento.

5.1. Etapa Zero

Inicialmente, os experimentos não foram realizados com equilíbrio de classes no conjunto de teste. Ao variar a taxa de aprendizagem, ocorreu o gráfico da Figura 2:

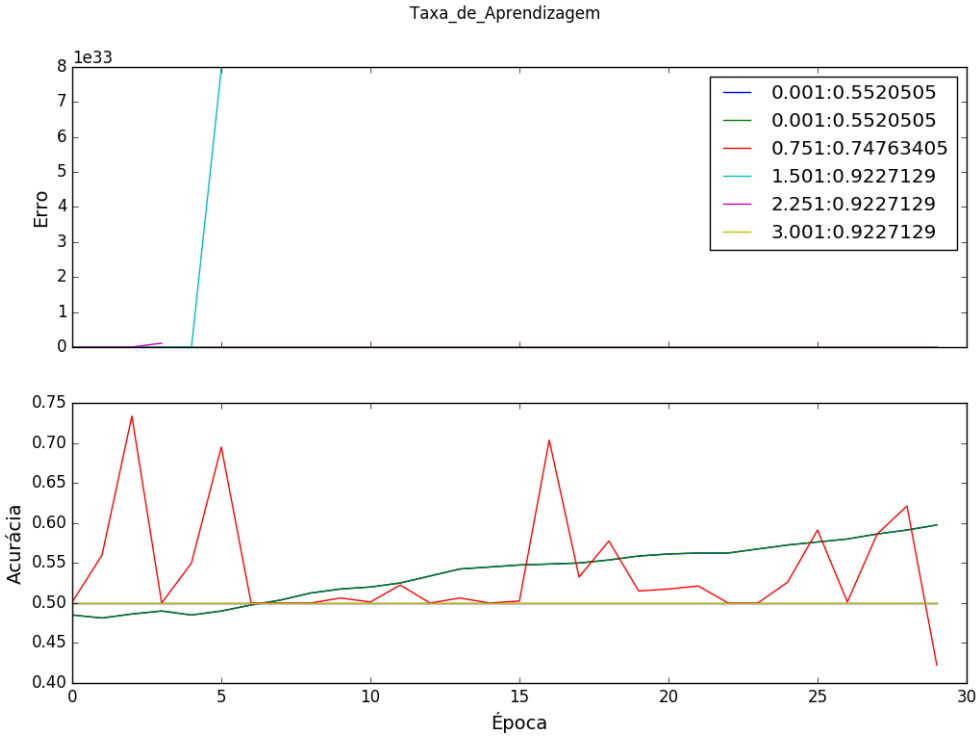

Figura 2.

Neste gráfico, é possível observar que o erro absoluto médio da curva 1.501 é alto no conjunto de treino. Sua acurácia, assim como as curvas 2.251 e 3.001, é de 50% para o conjunto de treino e 92,27% para o conjunto de teste. Isto torna claro que o modelo está prevendo sempre positivo. Para melhorar a medida da acurácia no conjunto de testes, este também foi equilibrado para comentários positivos e negativos.

5.2. Primeira Etapa

Nesta etapa, os gráficos representam acurácia e erro do modelo em relação ao conjunto de treino.

5.2.1. Pesos Iniciais (Valores Próximos)

O gráfico resultante do treino de 7 redes é apresentado a seguir, pela Figura 3 e o gráfico de acurácia dos 20 treinos pela Figura 4

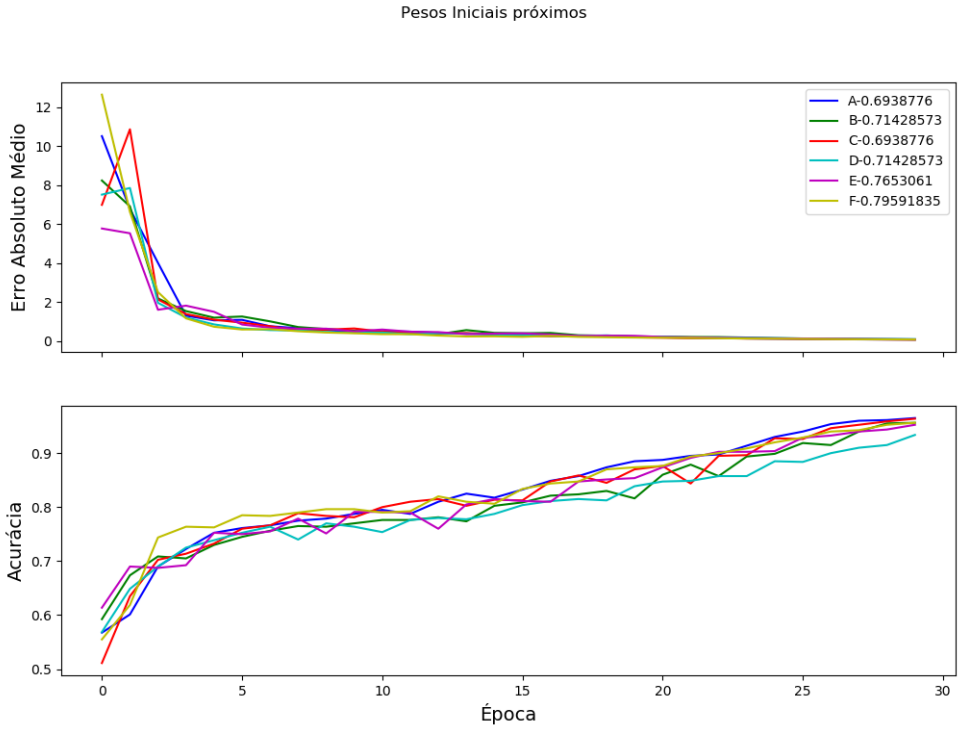

Figura 3.

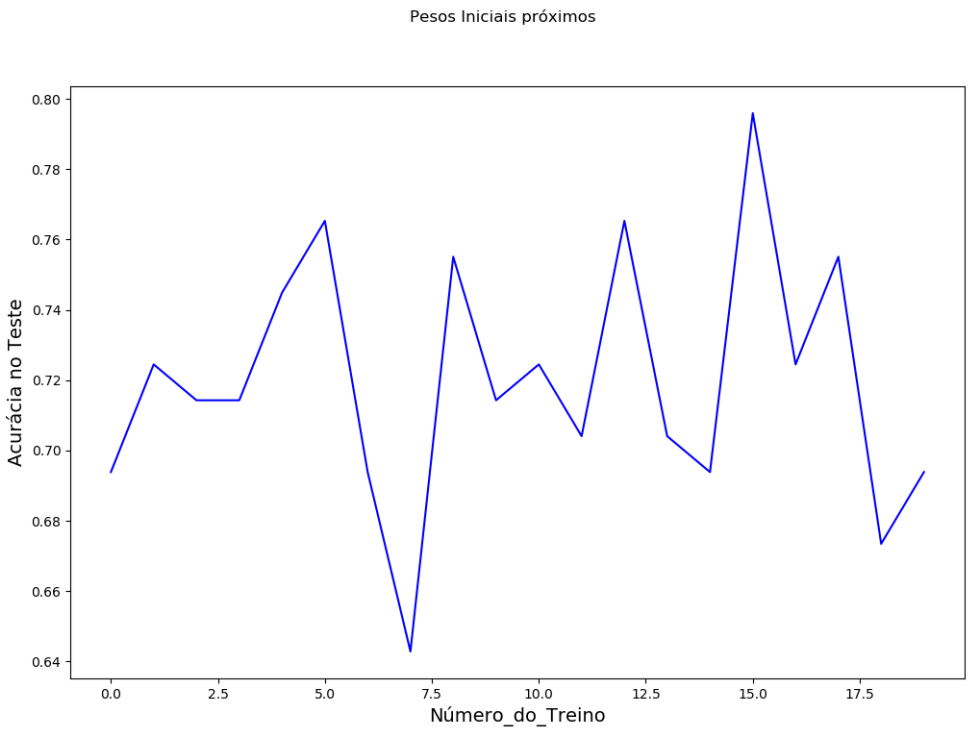

Figura 4.

Na Figura 3, o comportamento de todas as curvas é similar, tanto para acurácia quanto para o erro. A acurácia aumenta bastante no início do treino e aumenta cada vez menos ao longo das épocas. O erro em relação ao conjunto de teste cai rapidamente no começo do treino e, próximo de 1, o erro quase estabiliza, caindo muito devagar. O comportamento similar é esperado, visto que as configurações das redes neurais são idênticas, excetuando-se os pesos inicias, que são similares.

As curvas C e D apresentam o estranho comportamento de aumentar o erro após a primeira época de treino. Este fato não foi explicado por este trabalho e permanece como um item a ser pesquisado.

Estas curvas demonstram dois pontos importantes. O primeiro ponto é que, mesmo com todas as configurações iguais, as redes neurais apresentam diferença de até 15,30% na acurácia devido a pequenas variações nos pesos iniciais, como visto na Figura 4. Como o modelo e os dados representam uma topologia de muitas dimensões, mesmo pequenas variações dentro de um pequeno sub-espaço vetorial ocasionam caminhos bastante diferentes para a função de otimização.

O segundo ponto é que a acurácia no conjunto de treino não é proporcional à acurácia no conjunto de teste. As curvas A e C possuem as maiores acurácias em relação ao conjunto de treino, muito próxima de 100%, porém a menor acurácia em relação ao conjunto de teste. Da mesma forma, a curva E possui a segunda menor acurácia em relação ao conjunto de treino, mas a segunda maior acurácia em relação ao conjunto de teste.

5.2.2. Pesos Inicias (Valores Distantes)

Da Figura 5, imediatamente dois comportamentos são claros: o erro absoluto médio cresce em ordem de magnitude muitas vezes maior que a linear a medida que os pesos crescem e a acurácia do modelo no conjunto de treino é mais instável a medida que os pesos crescem.

O crescimento do erro absoluto médio é esperado, uma vez que os primeiros testes são realizados com valores muito próximos de zero e os últimos testes com valores maiores que 1. Enquanto os primeiros valores do peso dividem o erro, os últimos valores o multiplicam. Outro fator importante é o número de dimensões, 1326. Com esta quantidade de dimensões, mesmo pequenas variações em cada valor ocasionam grandes distâncias entre os pontos, como também visto pelo experimento anterior.

O fato do erro absoluto médio ser maior implica na velocidade do treino. Como visto no gráfico, o erro absoluto médio de curvas como a 1.232 e 1.4525 demoram metade das épocas de treino para se equiparar com curvas de pesos iniciais menores.

A instabilidade da acurácia pode ser explicada pela dificuldade de treinamento da rede. Uma vez que os erros absolutos são muito altos, também são as correções dos pesos. A uma taxa de aprendizado relativamente alta, estes valores podem ser corrigidos, ocasionando acurácias de teste bastante próximas.

Ao final do treino, no entanto, a variação máxima de acurácia foi de 9,18%, menor do que a variação para pesos inicias próximos. As redes neurais tiveram tempo (épocas de

treino) e ”força”(taxa de aprendizagem) suficiente para contornar os altos pesos iniciais.

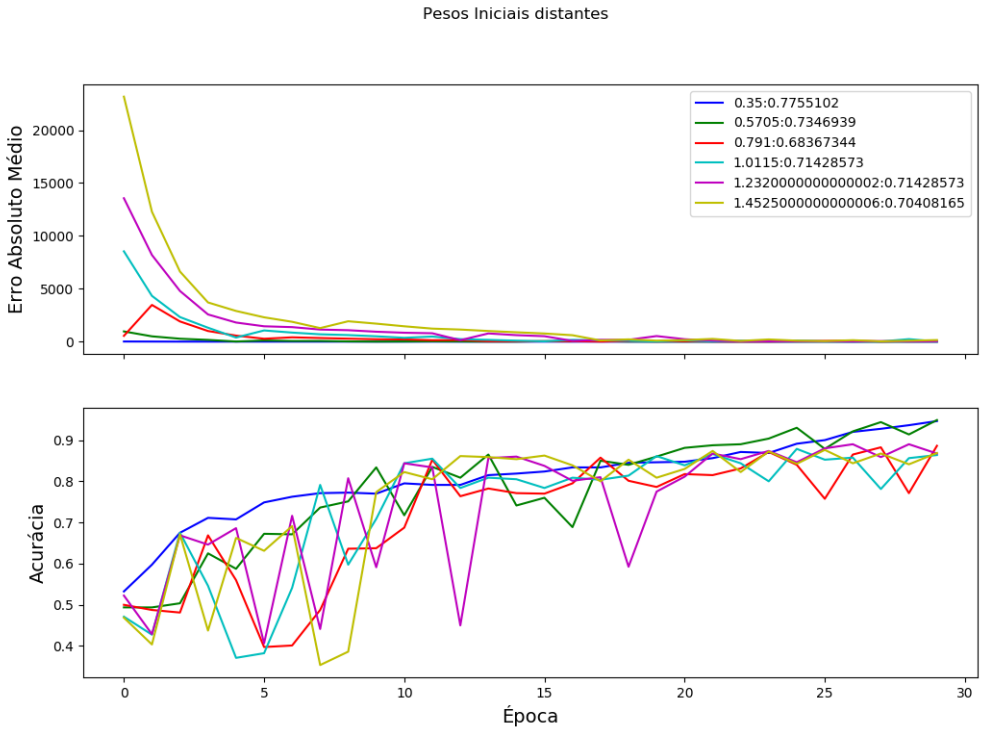

Figura 5.

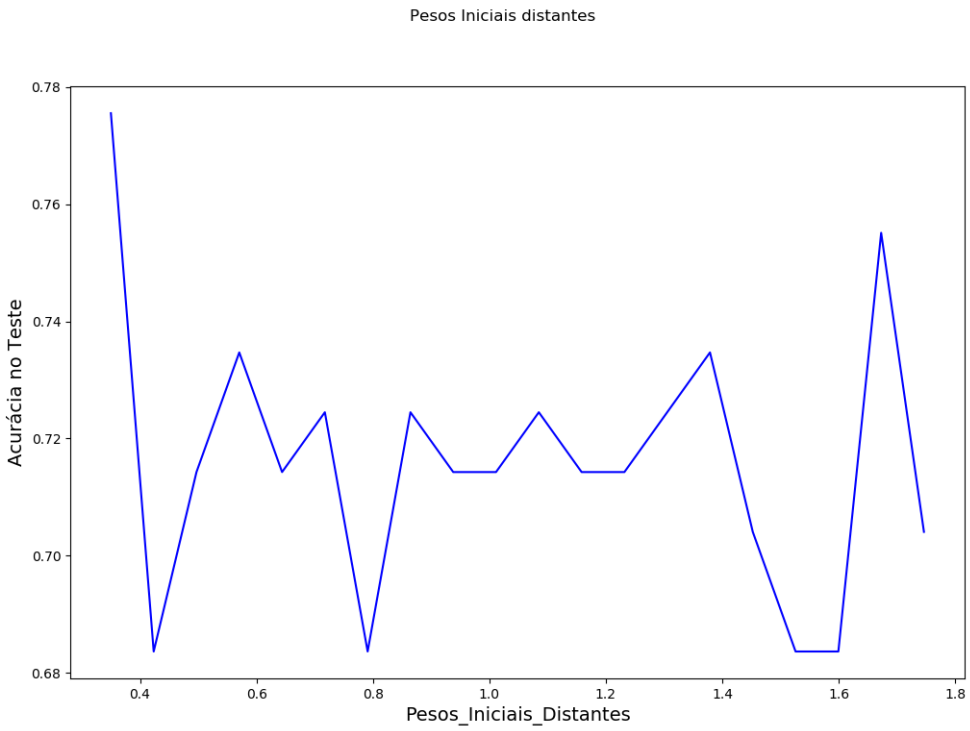

Figura 6.

No entanto, repetindo o experimento para taxas de aprendizado menores (Figura 7, fica claro o obstáculo imposto pelos pesos altos, necessitando muito mais épocas para serem treinados ou até inviabilizando o treino. Até mesmo para pesos pouco acima de 1 (curva 0.57), a instabilidade ainda é alta. Analogamente, como se as definições do que é positivo ou negativo fossem muito estabelecidas, dificilmente sendo quebradas, como um preconceito.

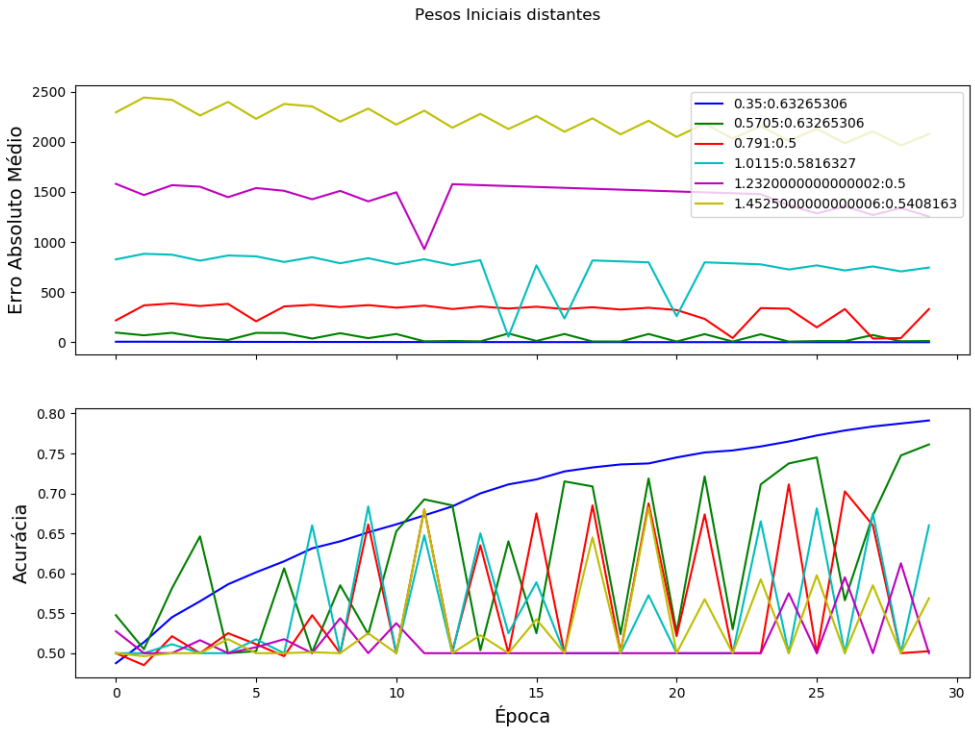

Figura 7.

5.2.3. Número de Neurônios

O aumento do número de neurônios também ocasiona erros absolutos médios maiores nas primeiras épocas de treino. Ao contrário dos grandes valores de pesos iniciais, no entanto, o erro se deve a pequenos erros absolutos de cada neurônio, que somados contribuem para um valor grande. Isto é visto pela magnitude do erro, não crescendo muito mais vezes que o número de neurônios.

Este crescimento também não impõe um obstáculo ao treino, uma vez que são todos pequenos erros. Isto é observado no gráfico: já na segunda época de treino os erros absolutos de todas as variações se encontram bastante próximos.

A inserção de mais neurônios parece causar turbulências na curva de acurácia. Isto pode ser explicado pela existência de mais valores contribuindo para o valor final, de forma que pequenos ajustes em cada valor ocasionam uma variação grande no valor final.

Ainda mais clara é a conclusão sobre o aprendizado final. Para número muito baixos de neurônios, a acurácia sobre o conjunto de treino não consegue crescer muito,

estabilizando-se próximo dos 80%. Enquanto que, para uma grande quantidade de neurônios, o valor se aproxima de 100%. Novamente, isto não reflete a acurácia final do modelo. A curva de 50 neurônios parece não ter aprendido o suficiente para uma boa predição, mas a curva de 200 neurônios prevê melhor o conjunto de testes do que as demais curvas, mesmo tendo menos neurônios e aprendido menos padrões do conjunto de treino.

O impacto da variação do número de neurônios na acurácia final não é avaliado mais profundamente, pois para variar o número de neurônios também se faz necessário variar os pesos inicias, o que já varia bastante a acurácia final. Desta forma, é complicado discernir qual o impacto do número de neurônios no resultado final.

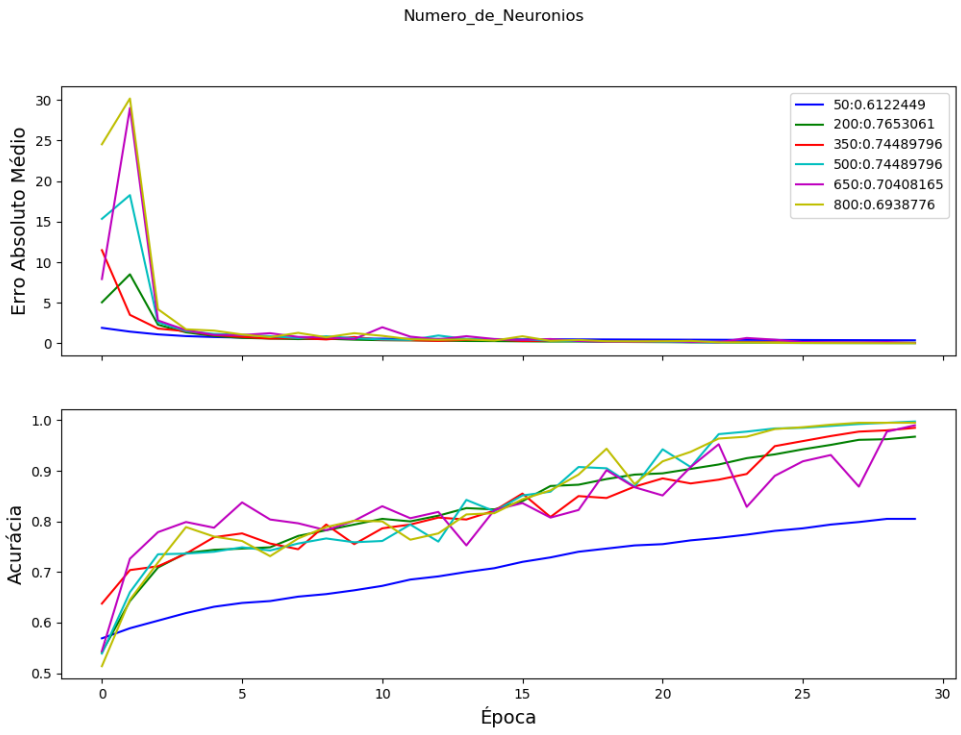

Figura 8.

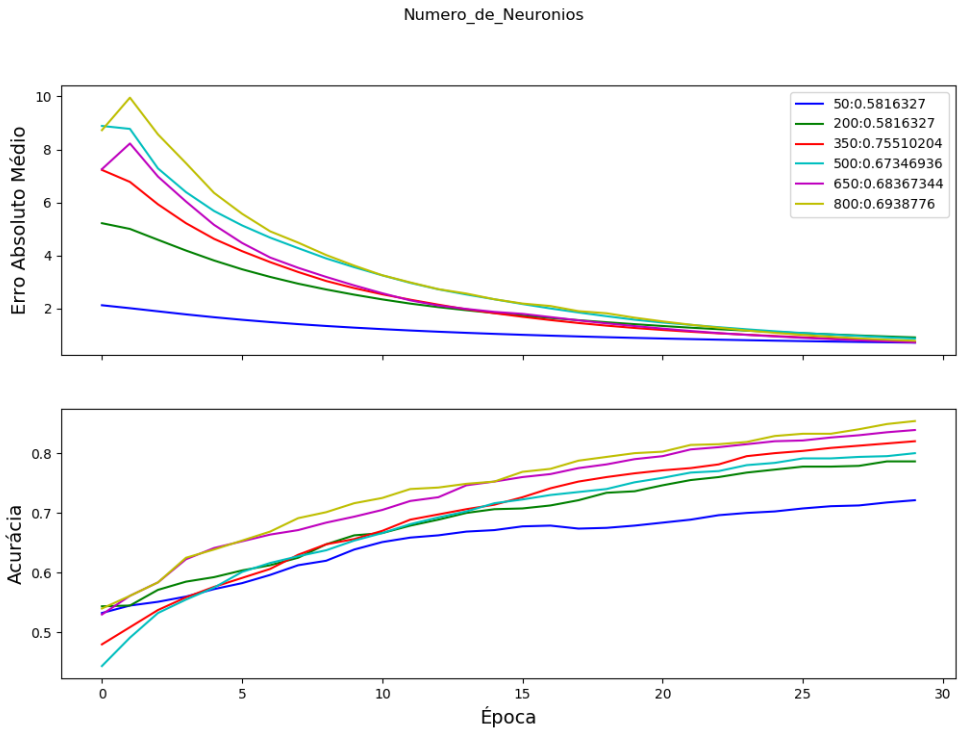

Figura 9.

5.2.4. Taxa de Aprendizagem

A Figura 10 demonstra os efeitos de grandes variações na taxa de aprendizagem. Para taxas pequenas (curva 0.001, azul) o aprendizado se da de forma pequena e gradual. Para taxas já muito maiores (curva 0.501, verde), o aprendizado é turbulento, porém ainda existe. Para taxas mais altas, não há aprendizado. O erro absoluto inclusive aumenta bruscamente, pois a alta taxa de aprendizado elevou ou reduziu bruscamente os pesos, criando uma situação similar a dos pesos iniciais distantes.

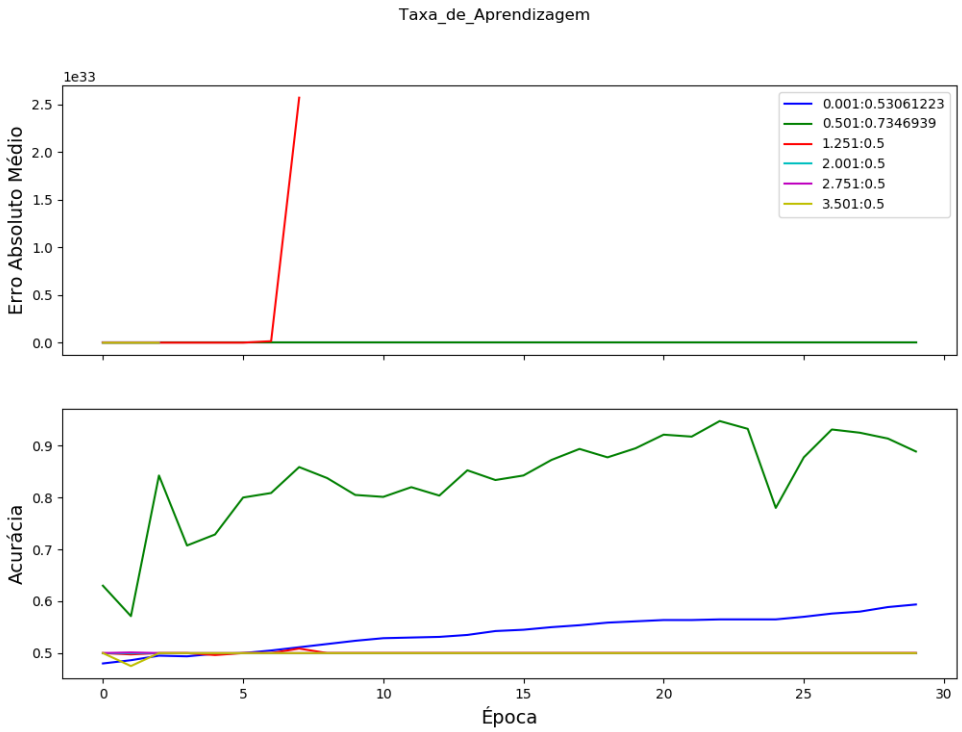

Figura 10.

As Figuras 11 e 12 apresentam variações mais sutis da taxa de aprendizagem (entre 0.01 e 0.40). Este comportamento parece mais intuitivo. Taxas de aprendizado maiores convergem mais rápido para o local ótimo. Taxas diferentes ocasionam em diferentes locais ótimos. Neste exemplo, a taxa mais acurada é a de 0,11, curva vermelha.

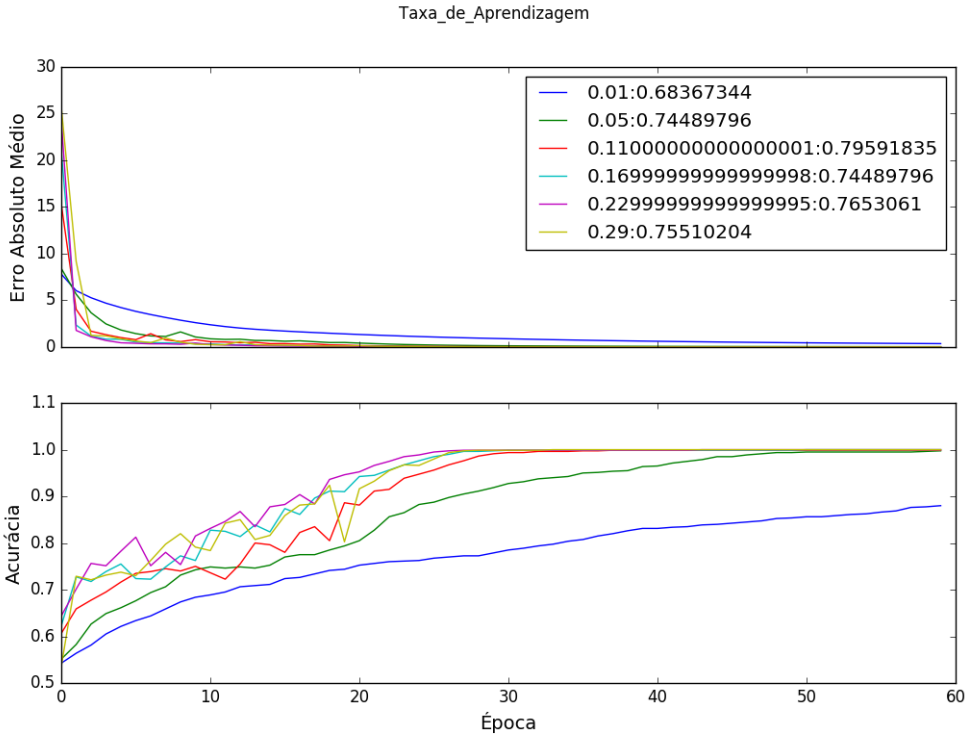

Figura 11.

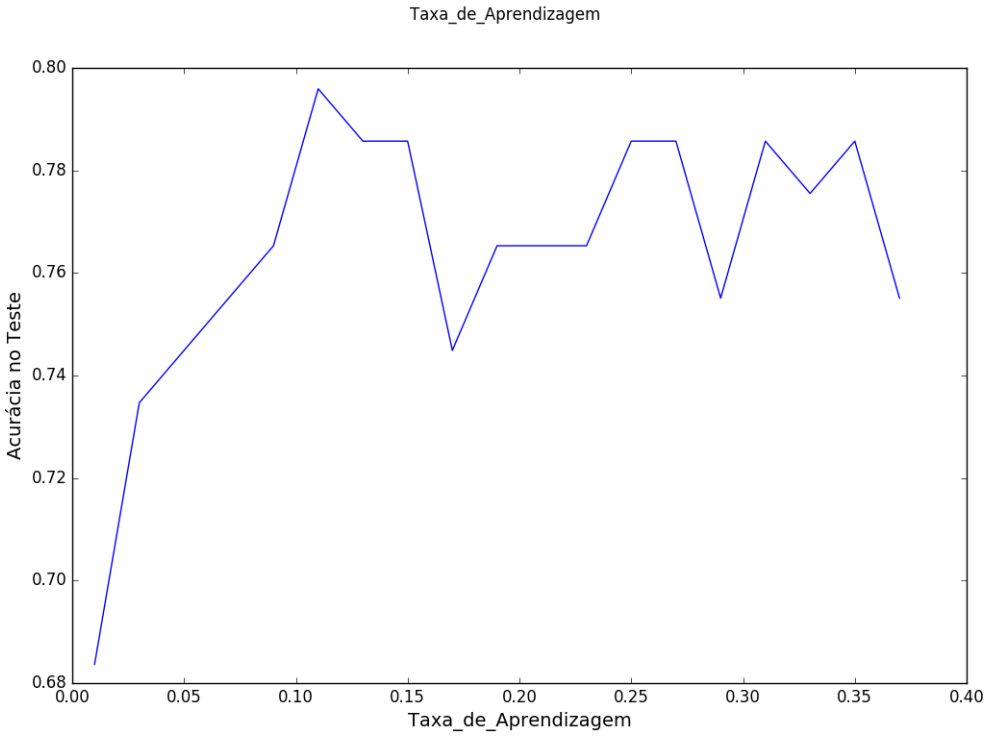

Figura 12.

5.2.5. Tamanho do *Batch*

A quantidade de comentários analisados antes do ajuste de pesos, ou tamanho do *batch*, também influencia a aprendizagem. Caso a rede analise cada comentário separadamente, não há aprendizado. Ou seja, se cada comentário aproximar ou afastar os pesos dele, a rede jamais chegará a uma direção resultante, sendo incapaz de aprender. Para *batches* pequenos, o aprendizado também é prejudicado. No entato, a partir de um determinado valor (neste caso 51 comentários por *batch*), o batch não possui mais influência no valor final da predição, mas parece influenciar a velocidade que o modelo chega a uma estabilidade.

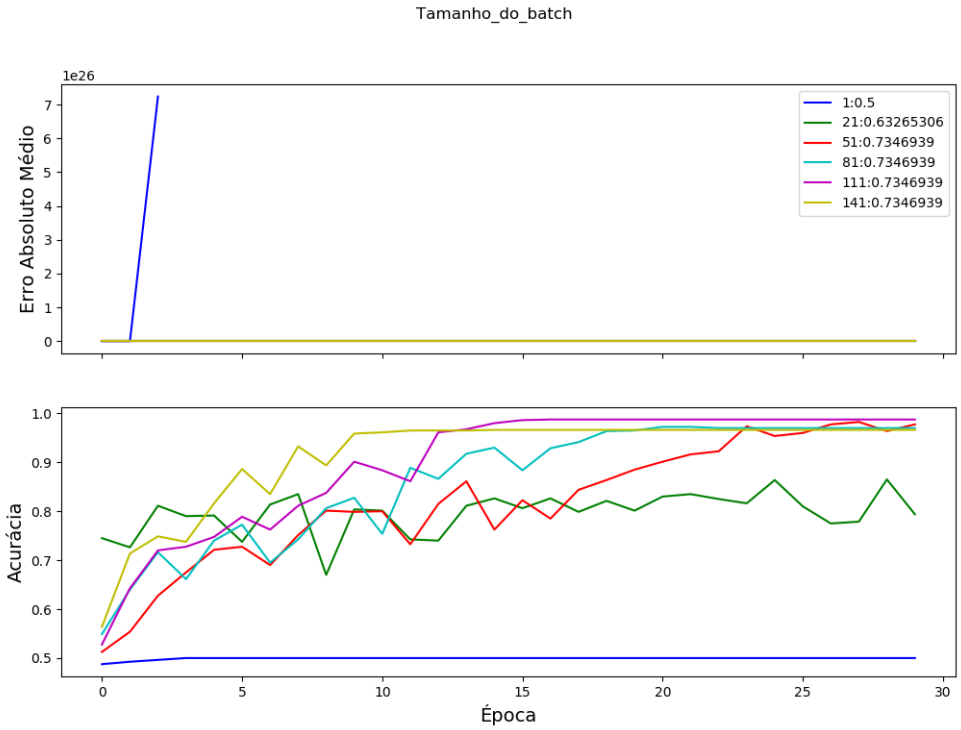

Figura 13.

5.2.6. Variação do Momento

A Figura demonstra a diferença entre variações pequenas e grandes de momento. Para momentos muito pequenos (curva 0.001 azul), o aprendizado se da com poucas turbulências, atingindo uma grande taxa de aprendizado no conjunto de treino. Para valores até 0.40 de momento, a rede ainda é capaz de aprender o conjunto de treino, porém com mais dificuldade. Para valores acima deste valor, a rede é incapaz de aprender, tomando inclusive o caminho oposto ao de aprendizado, com uma taxa de acerto menor que 50%. O erro anterior influencia cada vez mais a próxima otimização, ocasionando um erro ainda maior, que influenciará ainda mais a próxima medida. Desta forma, o erro cresce fora de controle.

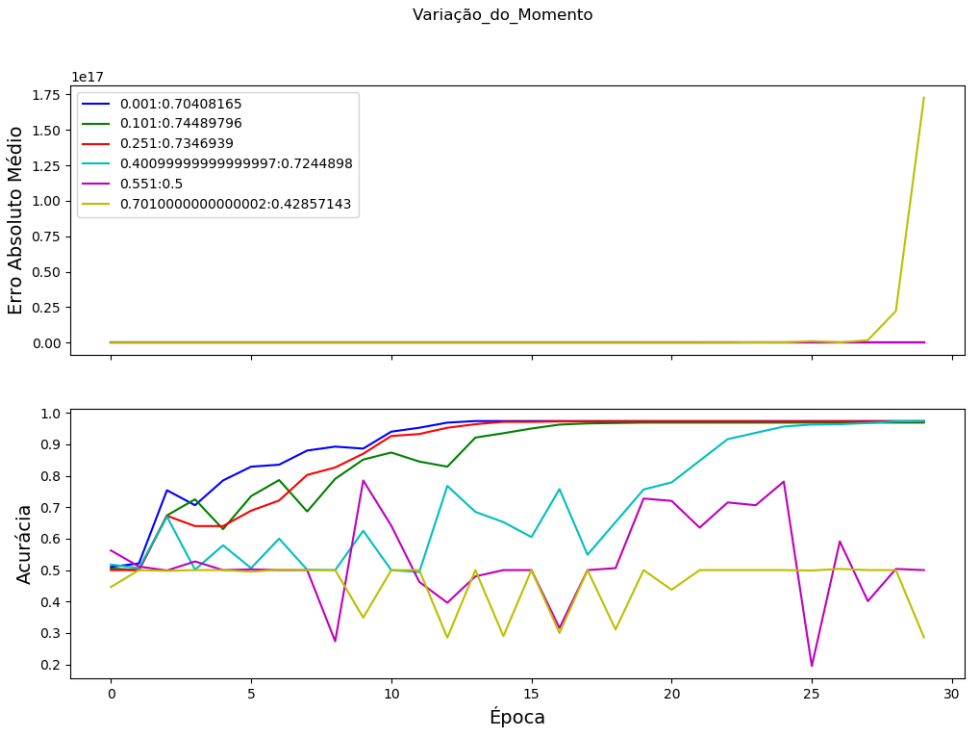

Figura 14.

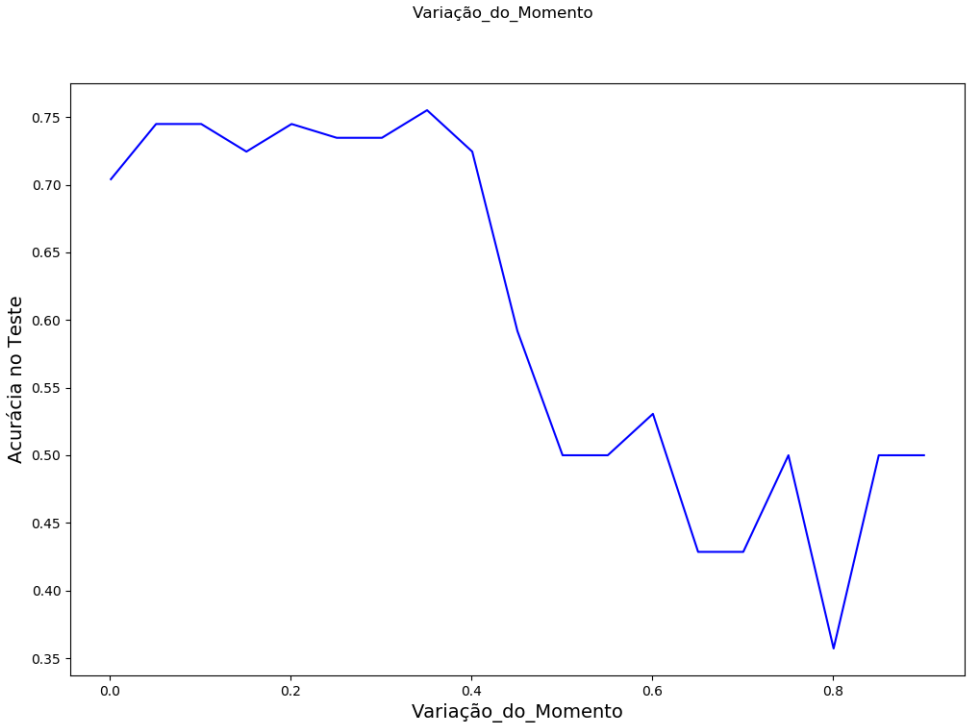

Figura 15.

5.3. Conjunto de Validação

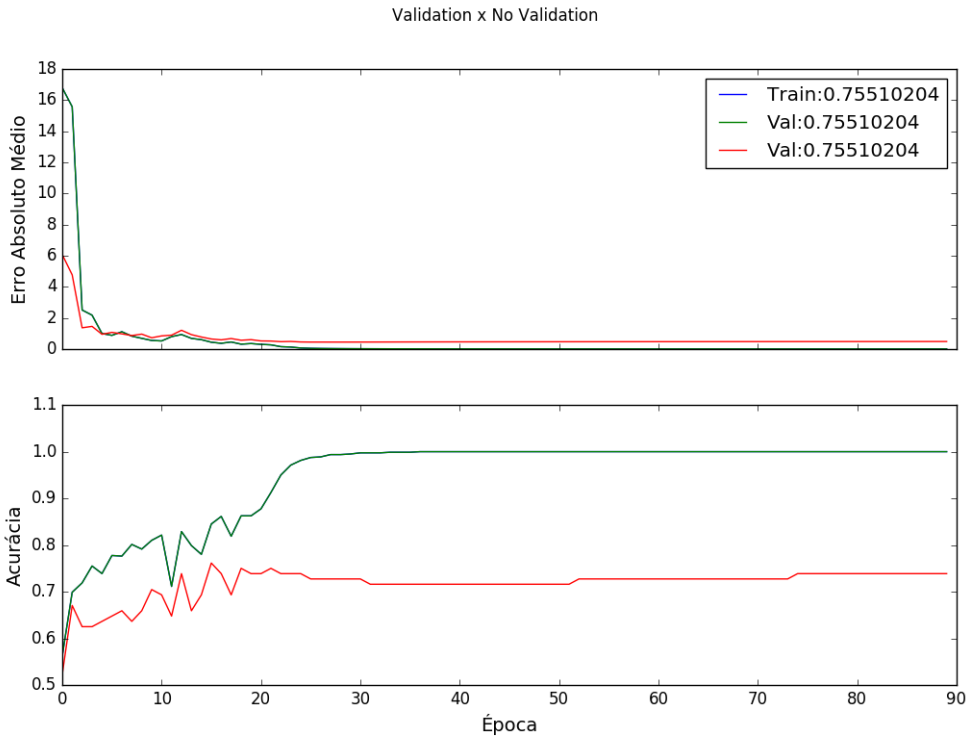

Figura 16.

A Figura 16 mostra a curva de acurácia do conjunto de teste e de validação para uma mesma configuração de rede. Embora a acurácia no conjunto de treino chegue a perto de 100%, a acurácia no conjunto de validação se estabiliza em um patamar. Este comportamento é o comportamento esperado do conjunto de teste. Por isso o conjunto de validação é implementado, para simular uma situação mais próxima à real.

6. Conclusão

Redes Neurais são complexas e o entendimento de seu funcionamento é difícil, mesmo uma Rede Neural Perceptron de apenas uma camada escondida. Muitos fatores influenciam na acurácia do modelo, além de influenciarem outros fatores. Isto é acrescentado a complexidade natural da modelagem do problema. Um modelo adequado requer bastante experimentação no problema específico. Algumas conclusões gerais de configuração, no entanto, parecem ser tiradas deste experimento, pois aparentam intuitivas matematicamente e são visualizadas nos experimentos.

O pré-processamento dos dados impacta consideravelmente o resultado do treino e da previsão. Pequenas variações de pesos inicias podem causar grandes variações na acurácia final. Não há relação direta de proporcionalidade entre acurácia no conjunto de treino e acurácia no conjunto de teste. Pesos inicias grandes, bem como alta taxa de aprendizagem ou momento inviabilizam o treino. Quantidades de elementos por *batch* muito pequenas ou número de neurônios na camada escondida muito pequenos também inviabilizam o treino.

Houveram fatores não estudados: a influência do conjunto de validação, o uso de diferentes otimizadores, número de comentários na base de dados, origem e distribuição (nos conjuntos) dos dados e diferentes representações para as palavras, como bag-of-words e skip-gram.