

- Atenção na concordância nominal

- Todas as figuras devem ser devidamente referenciada e descrita no texto

Relatório Técnico (50%)

Pad: Padrão (o trabalho contém todos os itens requeridos): 10,0

Ling: Linguagem (o texto está bem escrito, correto gramaticalmente e as ideias bem expressas): 8,0

Mod: Modelo (o modelo está bem descrito e sua instanciação feita adequadamente): 10,0

Res: Resultados (os resultados foram devidamente descritos, inclusive utilizando gráficos): 10,0

Aval: Avaliação (os resultados foram adequadamente apresentados): 10,0

Implementação (50%)

Com: Comentários: (o código está devidamente documentado): 10,0

Cod: Codificação: (o código está bem escrito e roda adequadamente): 10,0

Result: Resultados: (os resultados foram produzidos adequadamente, inclusive utilizando gráficos): 10,0

Nota: 9,5

**Relatório Técnico
sobre Redes Neurais Artificiais**

*Thiago M. de Sousa Luana G. B. Martins
Ruan C. Rodrigues*

Technical Report - RT-INF_000-19 - Relatório Técnico
July - 2019 - Julho

The contents of this document are the sole responsibility of the authors.
O conteúdo do presente documento é de única responsabilidade dos autores.

Relatório Técnico sobre Redes Neurais Artificiais

Thiago M. de Sousa
thiagomontelesofc@gmail.com

Luana G. B. Martins
luanagbmartins@gmail.com

Ruan C. Rodrigues
ruanchaves93@gmail.com

Abstract. *This report describes what were the decisions, solving process and results of the proposed problem in the field of Artificial Intelligence. The challenge of inducing a Artificial Neural Network that correlates with performance in the first two periods of the course with the final performance was introduced and allows to make predictions about the final performance of new students.*

Keywords: Technical Report, Neural Network.

Resumo. *Este relatório descreve quais foram as decisões, processo de resolução e resultados do problema proposto na matéria de Inteligência Artificial. Foi introduzido o desafio de induzir uma Rede Neural Artificial que tenha uma correlação com o desempenho nos dois primeiros períodos do curso com o desempenho final e permite fazer previsões sobre o desempenho final de novos alunos.*

Palavras-Chave: Relatório Técnico, Rede Neural Artificial.

1 Introdução

Este Relatório Técnico consiste na documentação de uma estratégia tomada para se obter um processo de tomada de decisões, buscando resolver o problema de se criar um modelo para obter previsões de desempenho final de novos alunos no Bacharelado em Ciência da Computação da Universidade Federal de Goiás, tendo em mãos apenas o histórico realizado nas matérias pertinentes aos dois primeiros períodos do curso.

O processo de decisão se deu por meio da utilização de uma rede neural artificial, as quais por sua vez tiram proveito de uma estrutura de dados com informações que levam o processo à sua conclusão através da simulação de aprendizagem de neurônios.

O problema consiste, em um primeiro momento, utilizar de dados de alunos que já concluíram o curso para definir parâmetros como período de ingresso, ano de conclusão e notas dos dois primeiros semestres para, em um segundo momento, criar um modelo que possa ser utilizado em uma Rede Neural Artificial, com o objetivo final de gerar previsões de desempenho de novos estudantes do curso com base em seu desempenho nos dois primeiros períodos.

No restante deste documento estão definidas a forma abordada na base de dados (Seção 2), descrição geral da solução proposta contendo nela a descrição geral do modelo utilizado e dos dados selecionados para o modelo (Seção 3), dos resultados obtidos (Seção 4), das propostas para como utilizar os resultados obtidos (Seção 5), conclusões finais (Seção 6) e referências.

gráficos?

2 Descrição da base de dados

Os dados consistem em um arquivo no formato csv (**Comma-separated values**) que é representado por uma matriz de 22361 linhas por 66 colunas, onde existe em cada coluna um determinado atributo referente a relação de um aluno com as disciplinas que cursou durante os anos e períodos.

	id	ano_nascimento_discente	idade_conclusao_ensino_medio	idade_ingresso_universidade	idade_colacao_grau	uf_naturalidade_discente
0	1	1989	17.0	19	26.0	GO
1	1	1989	17.0	19	26.0	GO
2	1	1989	17.0	19	26.0	GO
3	1	1989	17.0	19	26.0	GO
4	1	1989	17.0	19	26.0	GO

Os atributos nas colunas contém dados referentes ao aluno e sua passagem no curso. Dados como ano de nascimento, idade de ingresso à universidade e nota do Enem são exemplos de dados relacionados ao aluno anteriormente ao ingresso na faculdade. Já atributos como quantidade de reprovações, média global e ano de conclusão estão ligados ao aluno após a entrada na universidade.

Devido às restrições da definição de aluno de bom desempenho fornecidas pelo problema, alunos que ingressaram após o primeiro semestre de 2015 foram desconsiderados na etapa de treinamento.

3 Descrição da solução

Com o objetivo de permitir fazer previsões sobre o desempenho final de novos alunos, foram considerados alguns aspectos como a criação de uma classificação de bom aluno e o treinamento de um modelo utilizando disciplinas dos dois primeiros períodos do curso. Para o modelo foi proposto o uso de uma Rede Neural Perceptron Multicamadas (MLP, do inglês **Multilayer perceptron**).

3.1 Descrição do modelo utilizado

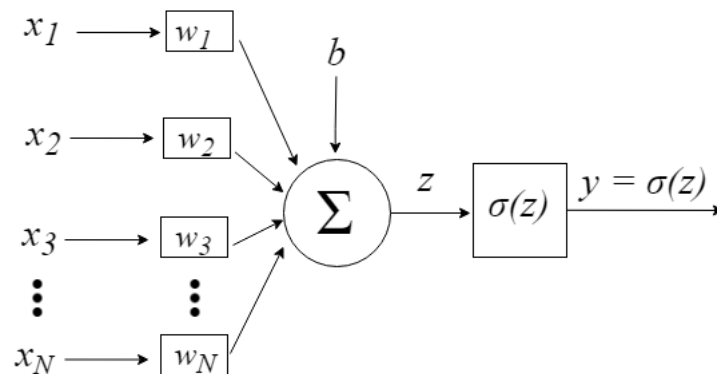
Nessa seção **será discutido os princípios** do funcionamento de uma MLP.

3.1.1 Estrutura do modelo de uma Rede Neural Perceptron

O modelo é inspirado no funcionamento de neurônios biológicos do sistema nervoso dos animais, ~~onde~~ foram feitas adaptações para que possa ser usado em um modelo computacional.

este modelo

Todas as figuras devem ser citadas no texto e devidamente explicadas



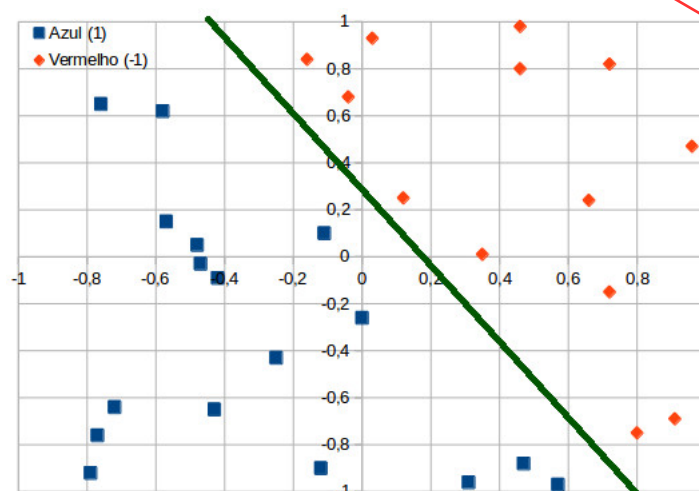
O modelo consiste em uma entrada de um vetor X contendo as entradas iniciais, representadas por $X = [x_1, x_2, x_3, x_n]$. No neurônio é adicionado um peso W_i a cada entrada X_i , gerando um valor Z , comumente denominado potencial de ativação, de acordo com a expressão abaixo:

$$Z = \sum_{i=1}^n x_i w_i + b \quad (-1)$$

O termo b corresponde a um **bias**, também conhecido como viés, sendo ela um número constante que pode ser interpretada como um conhecimento prévio auxiliar. O **valor** de Z é **passada** para uma função matemática de ativação σ , que corresponde a uma função que será responsável por criar um limiar do valor, que definirá a ativação (ou não) do dado neurônio, produzindo um valor de saída y do neurônio. [3]

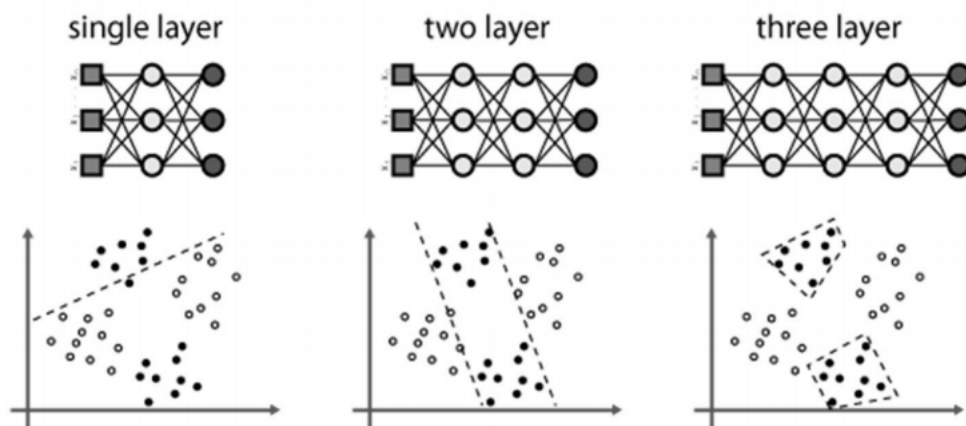
3.1.2 Rede Perceptron e Perceptron Multilayer

A rede Perceptron foi idealizada por Frank Rosenblatt em 1985. Esse algoritmo de aprendizagem supervisionada considera um período de treinamento com valor de entrada e saída que define se uma nova entrada pertence a alguma classe específica ou não. Sua arquitetura é composta de apenas uma camada de neurônios, e demonstrou uma boa acurácia para problemas de classificação. Entretanto, sua acurácia está restrita à problemas que sejam linearmente separáveis, ou seja, problemas dos quais é possível encontrar um hiperplano que seja capaz de realizar a separação do conjunto de dados, conforme mostra a figura abaixo:



Numerar e rotular as figuras

Para problemas não linearmente separáveis é preciso adicionar novas camadas contendo novos neurônios, constituindo uma MLP.



As MLPs apresentam uma ou mais camadas intermediárias de neurônios e uma camada de saída, onde as camadas intermediárias utilizam funções de ativação não lineares. ^No problema que foi nos proposto foi utilizado ^aa função sigmoideal que é a mais habitual em MLPs.

Em uma MLP, cada neurônio realiza uma função específica, onde a função implementada por um determinado neurônio em uma determinada camada é uma combinação das funções realizadas pelos neurônios da camada anterior que estão conectados a ele.

Na primeira camada, cada neurônio deduz uma função que define um hiperplano que divide o espaço de entradas em duas partes. Os neurônios da camada seguinte combinam um subconjunto das regiões convexas como uma combinação de retas. Os neurônios das camadas posteriores irão refinar as regiões, formulando funções que criam delimitações de uma forma mais específica, podendo assim criar fronteiras bem delimitadas das classes e permitir a classificação de novos exemplos.[1]

Para o problema proposto foi utilizado um modelo sequencial de camadas, onde a saída da primeira camada serve como entrada da segunda, e assim por diante, por ser o mais indicado para problemas de classificação.

3.1.3 Back-Propagation

Para realizar o treinamento de uma MLP é comum o uso de uma estratégia chamada de **Back-Propagation** ~~onde é feita~~ em duas partes:

- Fase **Forward**: Durante essa fase, cada objeto é apresentado à rede. Ele é recebido pelo neurônio da primeira camada intermediária da rede, onde é ponderado pelo peso associado a suas conexões. Cada neurônio na camada corrente é aplicada ^aa função de ativação (nesse caso, função sigmoide), essa função produz um valor de saída, que é utilizado como valor de entrada dos neurônios da camada posterior. Esse processo persiste até que os neurônios da camada de saída produzam cada um seu valor de saída, ~~onde~~ no final é comparado ao valor de saída desejado ~~a esse~~ neurônio. A diferença entre os valores de saída em relação ao valor de saída desejado indica o erro cometido pela rede para o objeto apresentado.

$$e_i = d_i - y_i \quad (-2)$$

- Fase **Backward**: Durante essa segunda fase, o erro calculado e_i , é propagado na rede no sentido inverso, em direção às camadas intermediárias, é nessa fase que ~~é realizado~~ os ajustes que serão aplicados ao parâmetros de modo que a rede obtenha um minimização do erro obtido. [2]

3.2 Definição final do modelo

Para o **treinamento do problema** foi utilizado um **dataset** com 9 colunas e 643 linhas, das nove colunas, oito serão referente aos atributos e uma classificação dos dados.

A Rede Neural Artificial será do tipo *feed-forward*, que consiste em:

- Camada de entrada: O modelo irá receber como entrada arrays de tamanho **(, 8)**, correspondentes aos atributos $X_{a,m}$ e $Y_{a,m}$ de cada aluno. A entrada será normalizada e escalonada.
- Camada intermediária: Cada neurônio recebe a entrada de tamanho **(, 8)** e é gerada uma saída de tamanho **(, 64)** a partir de 64 neurônios artificiais.
- Camada **dropout**: Elimina um conjunto aleatório de ativações na camada setado para zero. A principal motivação é dar maior robustez à rede para previsões fora da amostra, buscando capturar informações populacionais ao invés de características amostrais. Sua saída será de tamanho **(, 64)**.
- Camada de saída: Recebe uma entrada **(, 64)** e produz uma saída **(, 1)** através de uma função de ativação sigmoidal. Consiste em um único Perceptron. Emite uma classificação binária (0 ou 1).

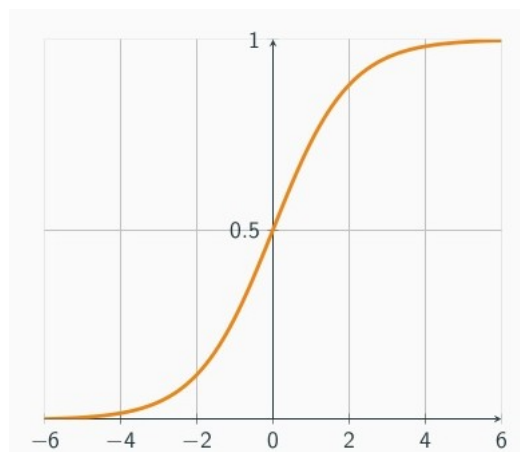
Onde as camadas serão densamente conectadas.

Na camada de entrada é utilizada uma função de ativação **relu** que é a unidade linear retificada. É necessária uma função de ativação que pareça e atue como uma função linear, mas é, na verdade, uma função não linear que permite relações complexas nos dados sejam aprendidas. A função é definida como:

$$f(x) = \max(0, x)$$

Já na camada de saída foi utilizada uma função de ativação sigmoidal. Esta é uma função suave e é continuamente diferenciável. Sua maior vantagem é a não linearidade: funções não-lineares permitem o aprendizado de estruturas mais complexas nos dados. A função varia de 0 a 1 tendo um formato de S, e é definida como:

$$f(x) = \frac{1}{1 + e^{-x}}$$



3.3 Descrição dos dados selecionados

De acordo com a definição do problema, aluno de bom desempenho é aquele que se forma em até 4 anos e meio, ou se forma com média igual ou superior a 7, ou se forma com número de reprovações igual inferior a 5.

Temos, portanto, a seguinte representação lógica para esta classificação:

$$A \vee (B \wedge C) \vee (B \wedge D) \quad (-3)$$

A: O aluno se forma em 4 anos e meio.

B: O aluno se formou.

C: O aluno tem média igual ou superior a 7.

D: O aluno tem número de reprovações igual ou inferior a 5.

Seja "bom aluno" um termo para o aluno que possui bom desempenho. Considere um aluno qualquer a e uma matéria m . A este par (a, m) iremos atribuir as variáveis $X_{a,m}$ e $Y_{a,m}$.

Seja $T_{m,n}$ a quantidade de bons alunos que passaram na matéria m na n -ésima tentativa, e U_m a quantidade total de bons alunos que já cursaram a matéria m . Então:

$$P_{m,n} = \frac{T_{m,n}}{U_m}$$

Seja $f_{a,m}$ uma função que retorna a quantidade de tentativas que o aluno a realizou para passar na matéria m .

Seja $g_{a,m}$ uma função booleana que retorna 1 caso o aluno a tenha sido aprovado na matéria m , e 0 caso ele tenha reprovado na matéria m mesmo após todas as suas tentativas. Então:

$$X_{a,m} = P_{m,f_{a,m}} * g_{a,m}$$

Seja $M_{m,n}$ a média geral dos bons alunos que cursaram a matéria m por n vezes.

Seja $N_{a,n}$ a nota do aluno a na n -ésima tentativa de passar na matéria.

Então:

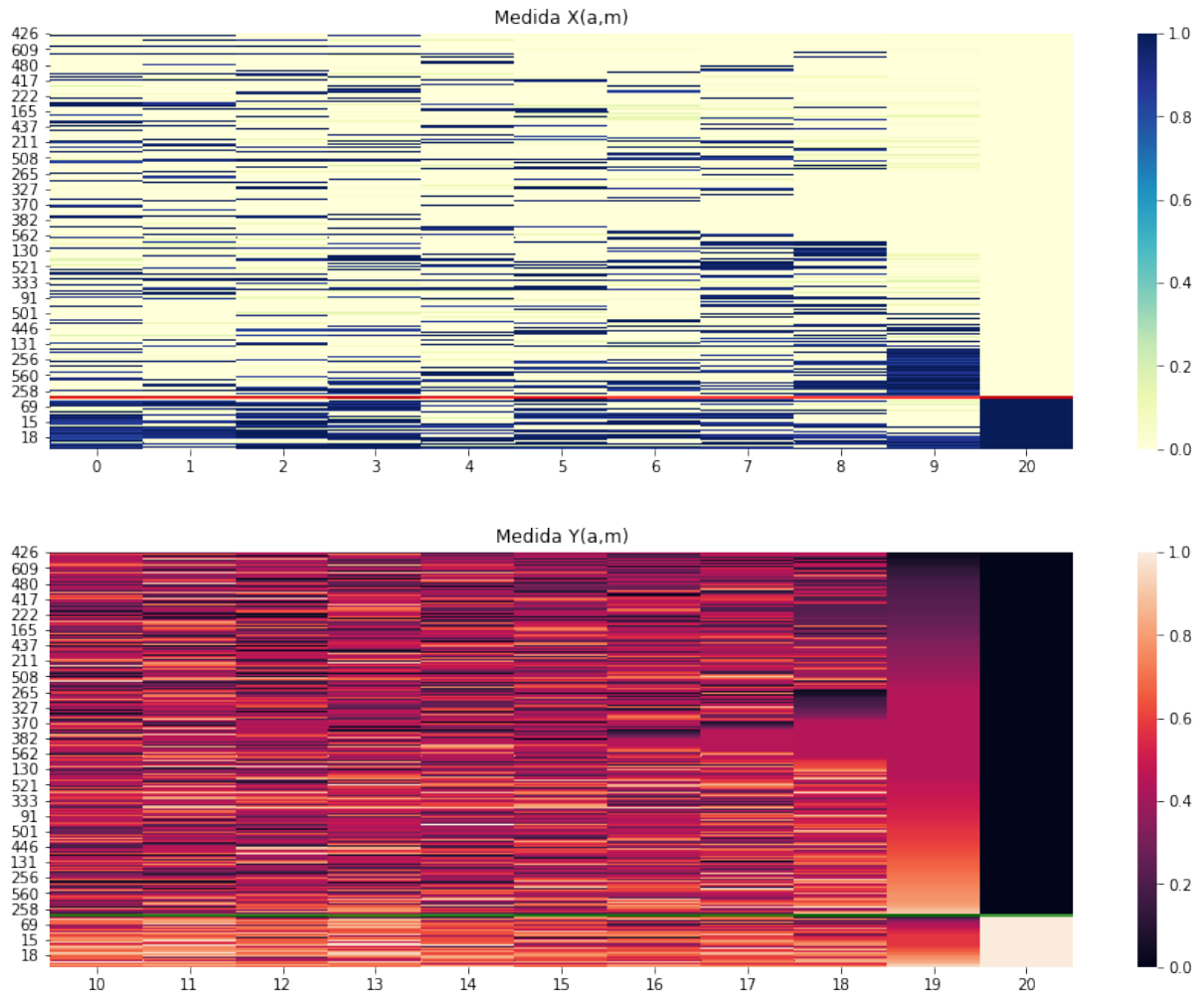
$$Y_{a,m} = \frac{\sum_{i=1}^{f(a,m)} P_{m,i} * D_{a,m,i}}{\sum_{i=1}^{f(a,m)} P_{m,i}}$$

Sendo $D_{a,m,n}$ o desvio normalizado:

$$D_{a,m,n} = \frac{(N_{a,n} - M_{m,n}) + 10}{20}$$

Após a inspeção visual dos dados através de recursos gráficos, foram selecionadas 4 matérias a serem introduzidas nos dados finais para o treinamento da árvore de decisão. As demais matérias foram desconsideradas.

A inspeção visual ocorreu através de *heatmaps*. Observe que as matérias desconsideradas apresentam aspectos muito semelhantes entre as regiões da coluna que estão acima e abaixo da reta que separa as categorias, enquanto as matérias consideradas apresentam colorações acen-tuadamente distintas entre as regiões acima e abaixo da reta.



Cada matéria contribui com um valor $X_{a,m}$ e um valor $Y_{a,m}$ para cada aluno a .

- Cálculo 1 (índices 0 e 10)
- Lógica Matemática (índices 1 e 11)
- Física para Computação (índices 3 e 13)
- Cálculo 2 (índices 5 e 15)

Os dados finais que serão considerados no treinamento consistem em um arquivo CSV contendo 643 linhas por 12 colunas.

	Unnamed: 0	0	1	2	3	5	10	11	12	13	15	20
0	0	0.114286	0.114286	0.861111	0.161765	0.000000	0.353357	0.353357	0.554274	0.363824	0.438167	1.0
1	1	1.000000	0.950820	1.000000	0.114286	0.138889	0.480515	0.477155	0.577063	0.385214	0.370347	1.0
2	2	0.161765	0.161765	0.845070	0.923077	0.969231	0.368750	0.368750	0.498167	0.483583	0.413968	1.0
3	3	0.114286	0.861111	0.161765	0.161765	0.923077	0.352357	0.494274	0.419706	0.419706	0.448583	1.0
4	4	1.000000	0.000000	0.861111	0.161765	0.845070	0.452063	0.403952	0.519274	0.313015	0.493167	1.0

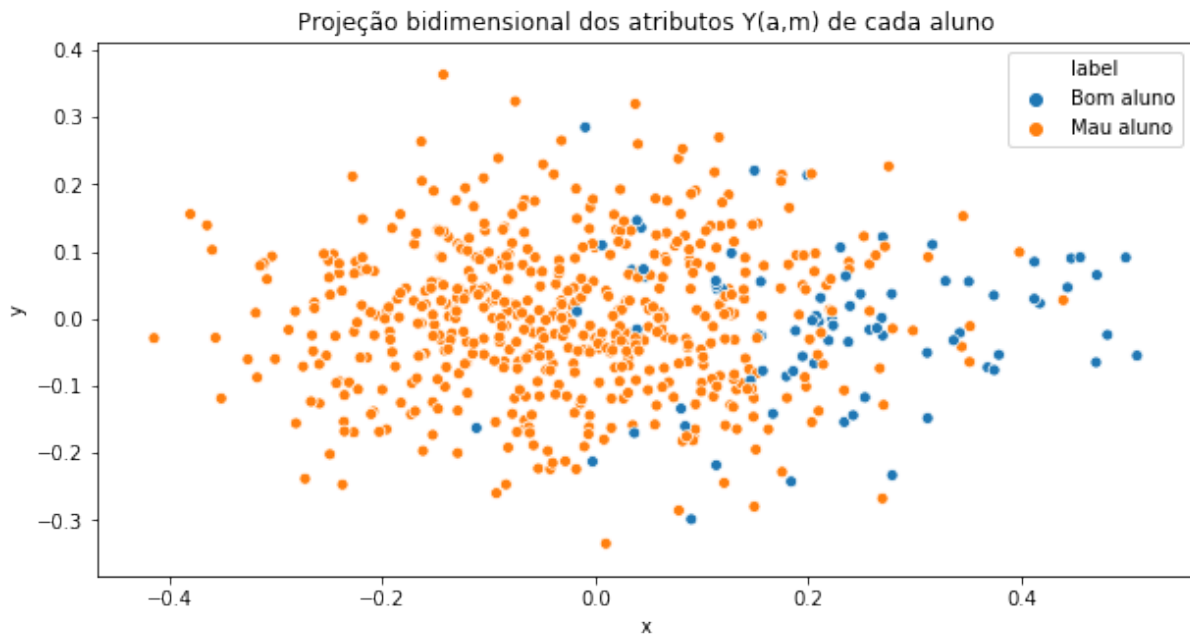
A primeira coluna representa uma identificação do aluno.

As colunas nomeadas de 0, 1, 2, 3, 5 representam o atributo $X_{a,m}$ de cinco matérias. Já as colunas 10, 11, 12, 13, 15 representam o atributo $Y_{a,m}$ destas mesmas cinco matérias. Atributos

referentes à mesma matéria estão separados por dez unidades: sendo assim, as colunas 0 e 10, 1 e 11, 2 e 12, 3 e 13 e 5 e 15 são referentes à mesma matéria.

A coluna nomeada 20 é a classificação final, que denota se o aluno pode ser classificado como aluno de bom desempenho (valor 1) ou não (valor 0).

Na etapa preliminar à elaboração de rede neural, realizamos uma visualização dos vetores obtidos através de um algoritmo de redução de dimensionalidade, o PCA (*Principal Component Analysis*). Realizamos uma projeção parcial de cada vetor, considerando apenas os valores $Y_{a,m}$.



A partir do PCA, formulamos as seguintes hipóteses:

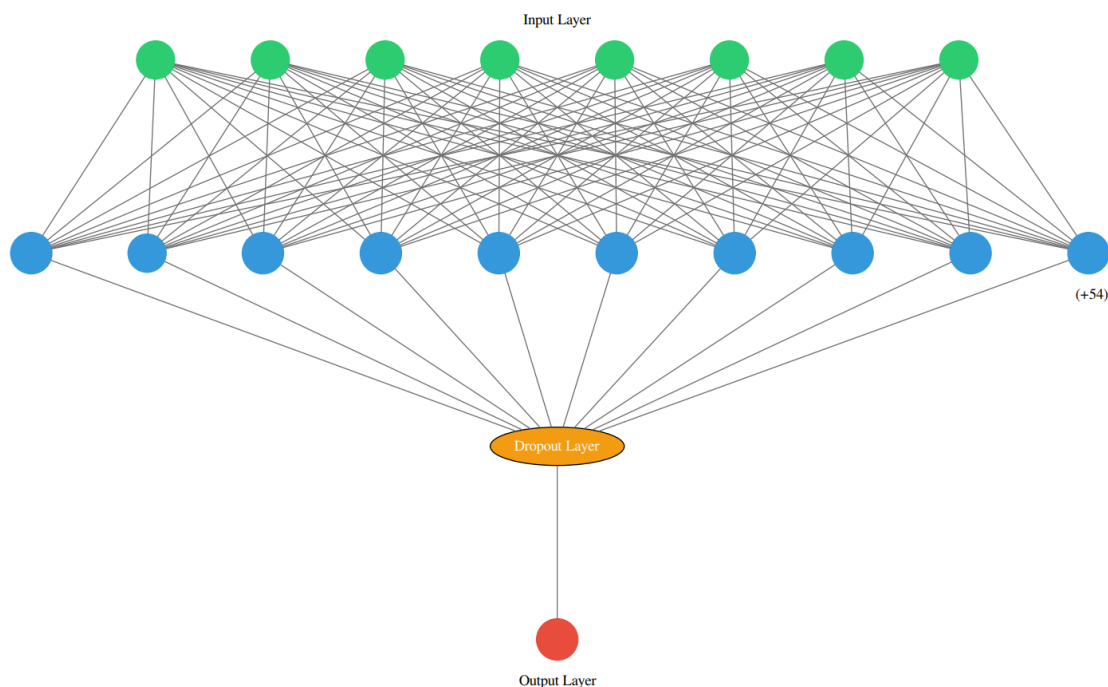
- Como $Y_{a,m}$ se aproxima de uma transformação linear de $X_{a,m}$, não haverá prejuízo da propriedade linearmente separável das classes caso a variável $X_{a,m}$ seja considerada no modelo. Portanto, podemos extrapolar as propriedades observadas na projeção parcial dos valores $Y_{a,m}$ para o vetor integral, considerando $X_{a,m}$ e $Y_{a,m}$ para cada matéria. Em outras palavras, teremos melhor acurácia se considerarmos tanto $X_{a,m}$ quanto $Y_{a,m}$ no treinamento do modelo.
- Um modelo MLP complexo, com várias camadas, terá performance equivalente a modelos simples com apenas uma camada intermediária, dado que as classes são aproximadamente linearmente separáveis.

Mais tarde, no decorrer do trabalho, estas hipóteses acabaram sendo confirmadas experimentalmente, através de testes, e chegou-se ao modelo ótimo apresentado a seguir.

4 Resultados obtidos

4.1 Rede Neural gerada

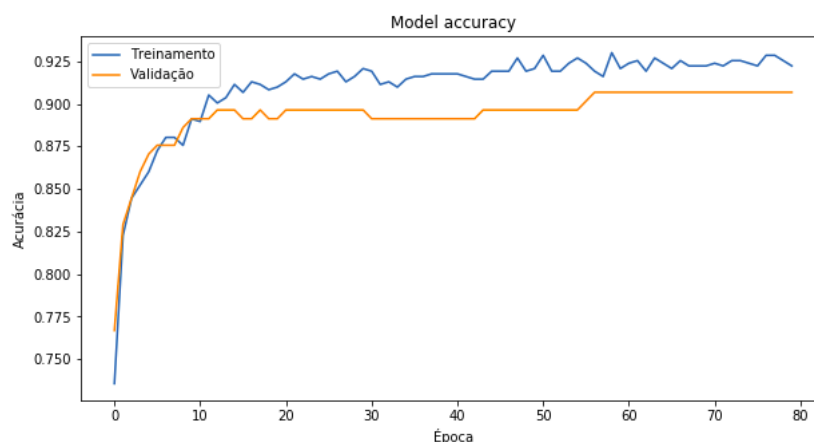
No final do processo foi gerada uma representação visual simplificada da rede neural construída.



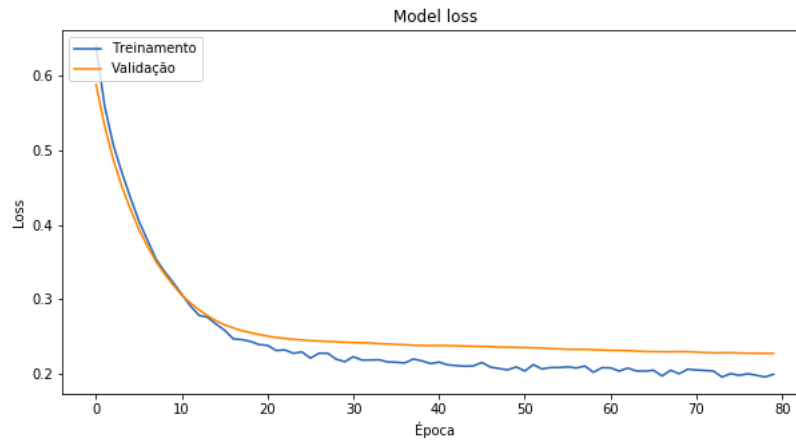
Em verde, temos as entradas (os valores $X_{a,m}$ e $Y_{a,m}$, escalonados e normalizados, para cada par de matéria/aluno a, m). Em azul, temos a camada intermediária, com 64 neurônios artificiais. Em laranja, temos a camada de Dropout. E em vermelho, temos o único Perceptron que recebe a saída da camada de Dropout e emite a classificação final do aluno através de uma função de ativação sigmoideal. Esta é a camada de saída do modelo.

4.2 Acurácia

O método `fit()` em um modelo Keras retorna um objeto `History`. O atributo `History.history` é um dicionário que registra valores de perda de treinamento e valores de métricas em épocas sucessivas, bem como valores de perda de validação e valores de métricas de validação (se aplicável). Com isso, conseguimos gerar um gráfico que mostra o comportamento da rede durante o treinamento.



O gráfico acima cria uma representação dos valores de acurácia conforme o treinamento e validação da rede.



O segundo gráfico mostra a perda do treinamento e validação conforme o algoritmo foi treinado.

	acc	pos	neg
mean	0.91	0.74	0.93
std	0.02	0.11	0.02

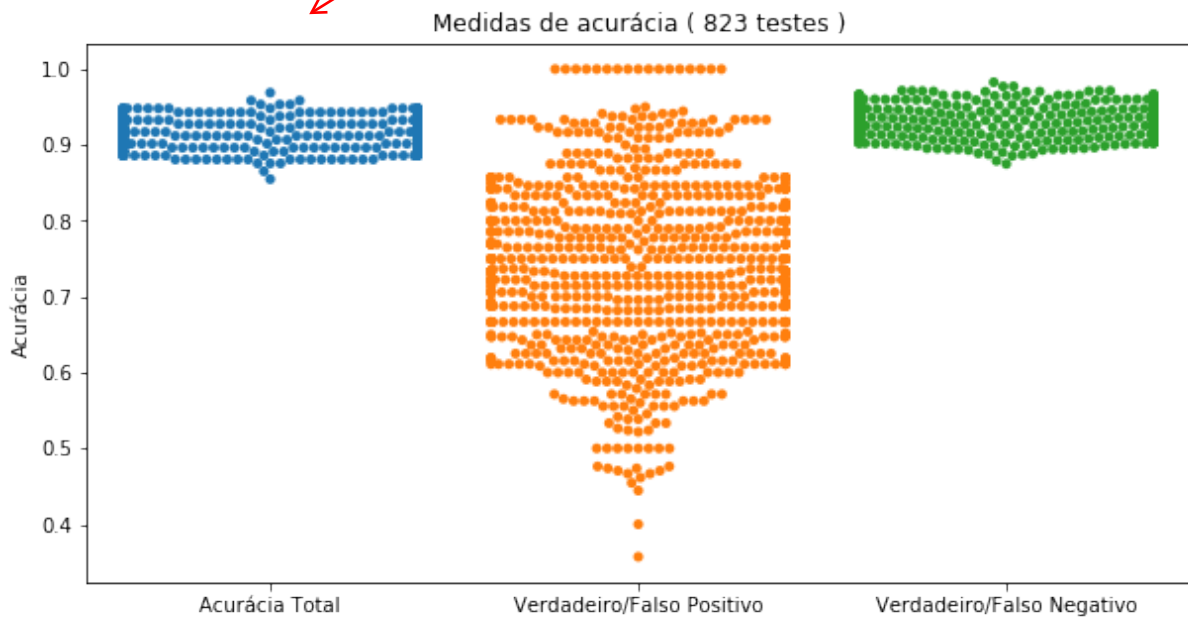
Padronizar as fontes das figuras

A tabela acima representa os valores de acurácia obtidos após a realização de 823 testes do modelo.

O modelo foi capaz de distinguir entre:

- (acc) Acurácia total de 93%, com erro médio de 2% para mais ou para menos.
- (pos) Distinção entre verdadeiro e falso positivo com 74% de acurácia, com um erro médio de 11% para mais ou para menos.
- (neg) Distinção entre verdadeiro e falso negativo com 91% de acurácia, com um erro médio de 2% para mais ou para menos.

O *swarmplot* abaixo representa todos os valores de acurácia obtidos como pontos. Há, portanto, 823 pontos por coluna, cada um correspondente a um teste.



5 Como a solução proposta pôde resolver o problema

O modelo atingiu alta precisão na detecção de verdadeiros negativos, e obteve resultados satisfatórios na distinção entre verdadeiros e falsos positivos, principalmente dado o baixo volume de entradas consideradas (por volta de apenas 600 vetores).

Observa-se, portanto, que o modelo é capaz de apresentar respostas efetivas a quaisquer perguntas e problemas que digam respeito a discernir entre alunos de bom e mau desempenho, ou projetar o desempenho futuro de um aluno, com resultados de classificação substancialmente acima da acurácia de uma escolha aleatória.

6 Conclusões

Constatamos que o modelo pode apresentar desempenho ainda melhor caso um volume maior de dados seja fornecido, e também caso seja dedicado um maior tempo ao ajuste fino de seus parâmetros (tais como os parâmetros do otimizador, o viés das camadas, o **dropout**, e regularizadores e restrições de cada camada). Entretanto, conclui-se que o modelo proposto obteve um bom resultado e correspondeu satisfatoriamente aos requisitos do problema que foi apresentado ao grupo.

Referências

- [1] Faceli, K. **Inteligência Artificial.** In: *Uma Abordagem de Aprendizado de Máquina.* Publishing Press, 2011.
- [2] Haykin, S. **FEEDFORWARD NEURAL NETWORKS: AN INTRODUCTION.**
http://media.wiley.com/product_data/excerpt/19/04713491/0471349119.pdf, último acesso em Julho de 2019.
- [3] Leite, T. M. **Redes Neurais, Perceptron Multicamadas e o Algoritmo Backpropagation.** <https://medium.com/ensina-ai/redes-neurais-perceptron-multicamadas-e-o-algoritmo-backpropagation->
último acesso em Julho de 2019.