Instituto Nacional
de Estudos e Pesquisas
Educacionais Anísio Teixeira

Classificação de Texto de Itens do Enem

CGMEB | Daeb
Johanes Severo

Brasília (DF) | 10/10/2023

MINISTÉRIO DA EDUCAÇÃO
GOVERNO FEDERAL BRASIL UNIÃO E RECONSTRUÇÃO

Objetivo

Explorar a utilização de um modelo de *deep learning* para classificar um item elaborado para o Exame Nacional do Ensino Médio com base no texto do item e no texto das alternativas, dentro de quatro faixas de valores (classes) do parâmetro de discriminação, parâmetro de dificuldade e o parâmetro de acerto ao acaso, ou seja, é um problema de classificação de texto.

Fases do processo: workflow padrão

- **Coleta de dados:** itens das provas do Enem e os parâmetros dos itens com base nos microdados;
- **Análise e exploração dos dados:** correlações;
- **Processamento/Tratamento de dados**: a base com os vetores de texto do itens e as classes anotadas para utilização na preparação de dados para o modelo;
- **Preparação dos dados para os Modelos de Aprendizado de Máquina**: divisão de treino e teste;
- **Aplicação de modelos de Aprendizado de Máquina**: treino do modelo Rede Neural Convolucional para classificação de texto;
- **Avaliação dos modelos de Aprendizado de Máquina e Discussão dos Resultados**: análise as métricas de avaliação do modelo;

Fases do processo: coleta de dados

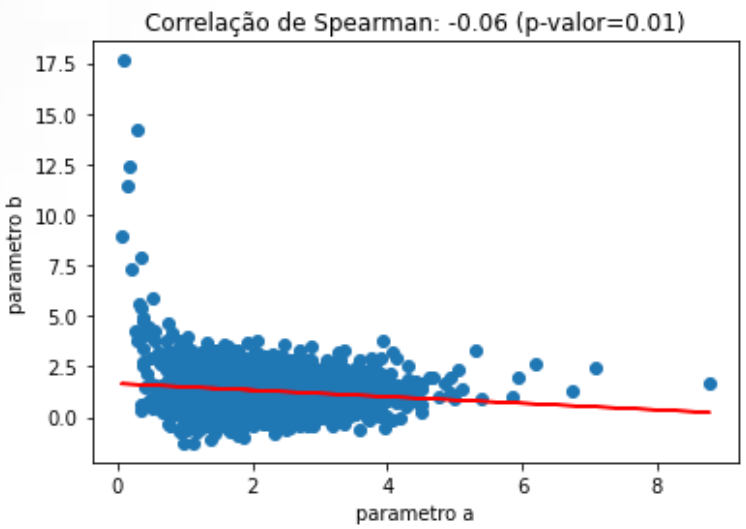
- Extração por meio de scripts python**;**
- **Textos**: pdf's nos microdados e sites;
- **Parâmetros**: TS_ITEM de todos os anos, csv's nos microdados;
- **Base de dados:** base associando a TS_ITEM com os textos extraídos das questões; Disponível em: https://github.com/johanessevero/classificacao_itens_enem_projeto_final_puc
- 1922 textos:

CH	487
CN	484
LC	442
MT	509

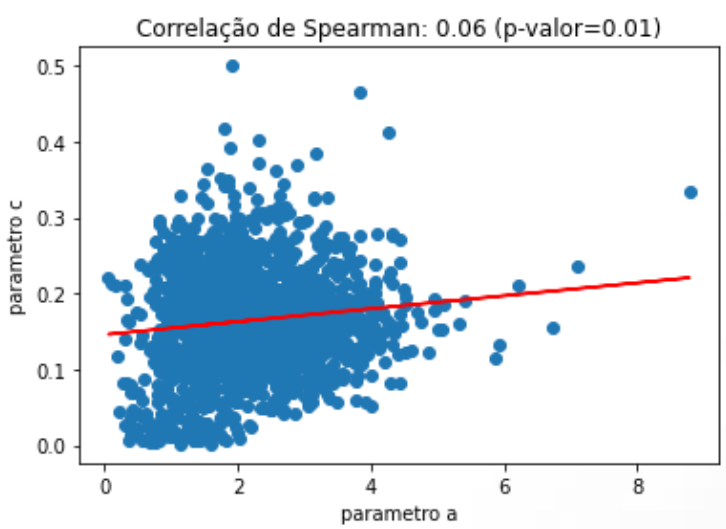
Fases do processo: **Análise e exploração dos dados**

- **Análise de correlação entre os parâmetros:** a hipótese é que se os parâmetros forem muito correlacionados os resultados dos modelos treinados serão semelhantes;

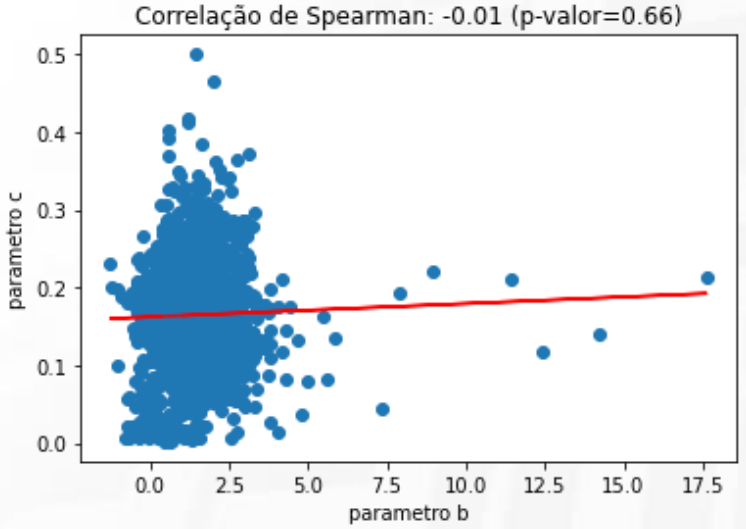
a x b



a x c



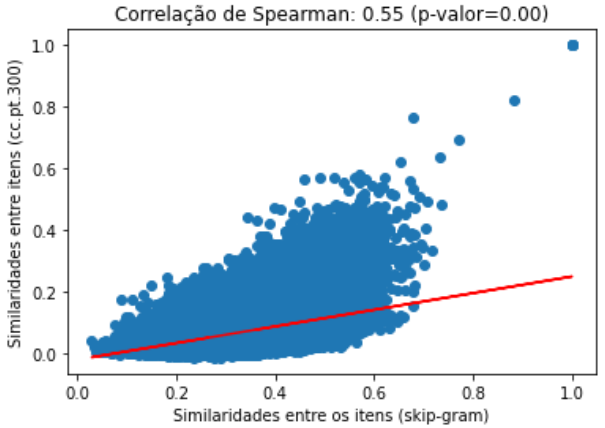
b x c



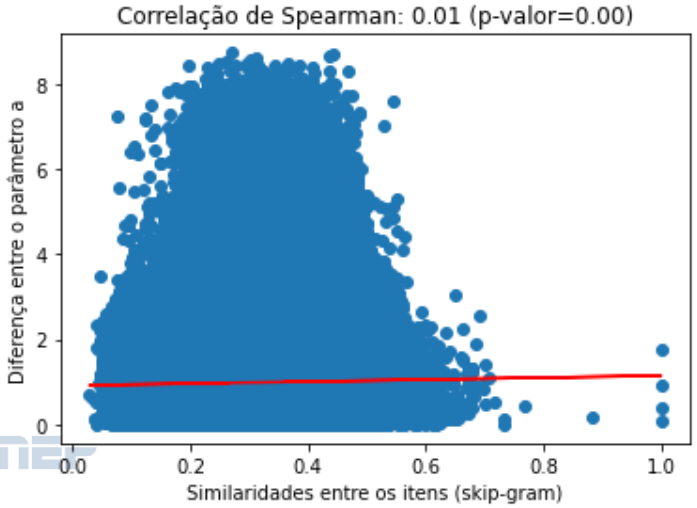
Fases do processo: Análise e exploração dos dados

- **Análise de correlação entre a matriz de similaridade item-item e a diferença absoluta entre os parâmetros dos itens:** a hipótese é de que, quanto maior a similaridade dos itens menor é a diferença entre os parâmetros;
  - Utilizou-se a representação skip-gram;

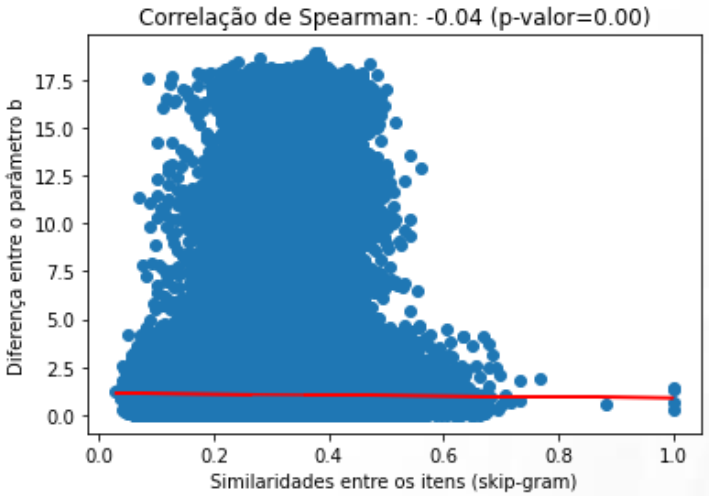
similaridade x similaridade



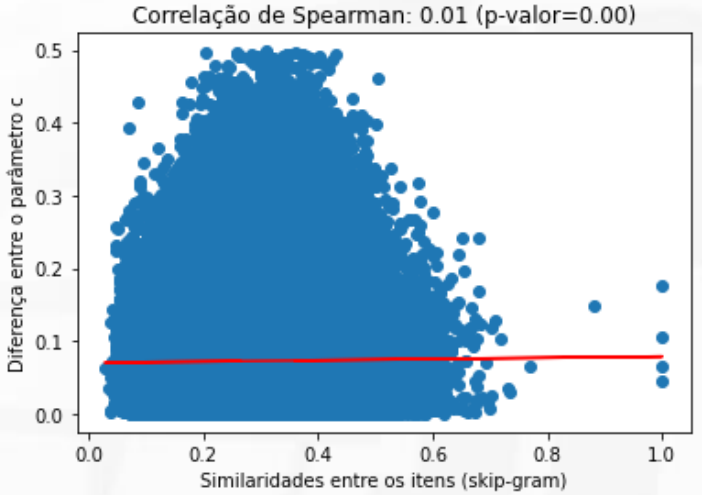
similaridade x a



similaridade x b



similaridade x c



Fases do processo: Tratamento de Dados

- **Divisão dos itens em classes de acordo com os quartis e representadas pelas faixas de valores dos parâmetros:**

classes_param_a	qtd
<=1.581	481
>1.581 e <=2.092	480
>2.092 e <=2.735	480
>2.735	481

classes_param_b	qtd
<=0.652	481
>0.652 e <=1.224	480
>1.224 e <=1.837	480
>1.837	481

classes_param_c	qtd
<=0.12487	482
>0.12487 e <=0.165	480
>0.165 e <=0.203	479
>0.203	481

Fases do processo: Tratamento de Dados

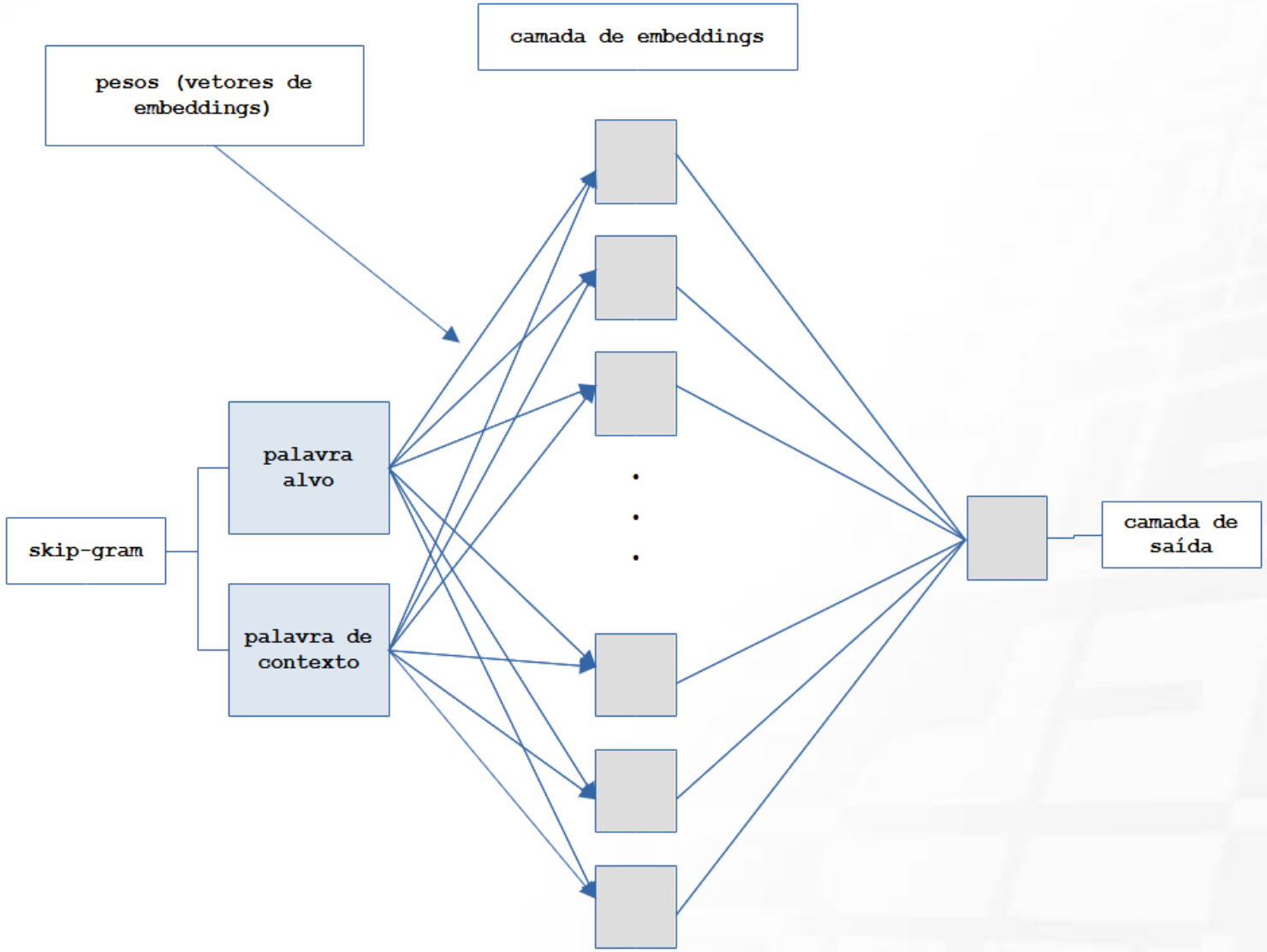
- **Limpeza do texto e obtenção das representações vetoriais (word embeddings) do texto:**
  - **Tokenização;**
  - **Remoção de stop words;**
  - **Lematização:** reduzir uma palavra à sua forma base ou forma canônica**;**
  - **Treinamento do modelo skip-gram:** palavras representadas por vetores de floats**;**
  - **Ex.: Palavra "concentrar" no contexto do corpus de questões do Enem:**

```
array([ -0.081, 0.084, 0.016, 0.116, -0.024, -0.135, 0.048,
0.249, 0.040, -0.039, -0.121, -0.122, -0.042, 0.038, -0.017, -
0.064, 0.057, -0.103, 0.037, -0.161, 0.021, -0.009, 0.048, -
0.025, 0.005, -0.013, -0.136, -0.119, -0.033, 0.052, 0.200,
0.076, -0.008, -0.012, -0.007, 0.088, 0.044, -0.130, -0.072, -
0.138, 0.044, -0.138, 0.038, -0.102, 0.080, -0.025, -0.054,
0.061, 0.017, 0.067, 0.154, -0.120, -0.022, 0.003, -0.099,
0.065, 0.111, -0.006, -0.103, -0.048, 0.030, 0.022, -0.013,
0.072, -0.104, -0.005, 0.013, 0.098, -0.124, 0.209, -0.070, -
0.038, 0.053, 0.003, 0.080, 0.028, -0.061, -0.026, -0.001, -
0.012, -0.071, -0.023, -0.095, 0.154, 0.024, 0.008, -0.033,
0.104, 0.185, 0.078, 0.066, 0.047, -0.038, 0.040, 0.183, 0.129,
-0.001, -0.143, 0.077, 0.033], dtype=float32)
```

  - **Texto tratado do item 1921:**

```
['escola', 'iniciar', 'processo', 'educativo', 'implantação', 'coleta', 'seletivo', 'material', 'reciclável', 'atingir',
'objetivo', 'instituição', 'sensibilizar', 'comunidade', 'escol', 'ar', 'desenvolver', 'atividade', 'sala', 'maneira',
'contínuo', '2', 'capacitar', 'pessoal', 'responsá', 'vel', 'limpeza', 'escola', 'quanto', 'adotar', 'coleta', 'seletivo', 'e3',
'distribuir', 'coletor', 'material', 'reciclável', 'específico', 'sala', 'pátio', 'acondicionamento', 'resíduo', 'completar',
'ação', 'proposta', 'ambiente', 'escolar', 'faltar', 'arealizar', 'campanha', 'educativo', 'sensibilização', 'bairro', 'vizinho',
'coleta', 'seletivo', 'parceria', 'prefeitura', 'cooperativa', 'catador', 'material', 'reciclável', 'destinação', 'apropriado',
'visita', 's', 'lixão', 'aterro', 'local', 'identificar', 'aspecto', 'disposição', 'final', 'lixo', 'rádio', 'local', 'jorna',
'l', 'impresso', 'rede', 'social', 'escola', 'coleta', 'seletivo', 'ecolocar', 'tes', 'coletor', 'lixo', 'reciclável', 'escola',
'população']
```

Treinamento de Vetores de Embeddings



Fases do processo: Preparação de dados

- Amostra estratificada por classe de 80% dados de treinamento 20% dados de teste;
- Dos 80% de dados de treinamento 20% são utilizados como dados de validação durante o treinamento do modelo;

classes_param_a	qtd total	qtd de trein. 1	qtd de teste	qtd de validação	qtd de trein. 2
<=1.581	481	382	99	76	306

Fases do processo: Treinamento do modelo de Deep Learning

Utilizou-se uma Rede Neural Convolucional com a seguinte arquitetura:

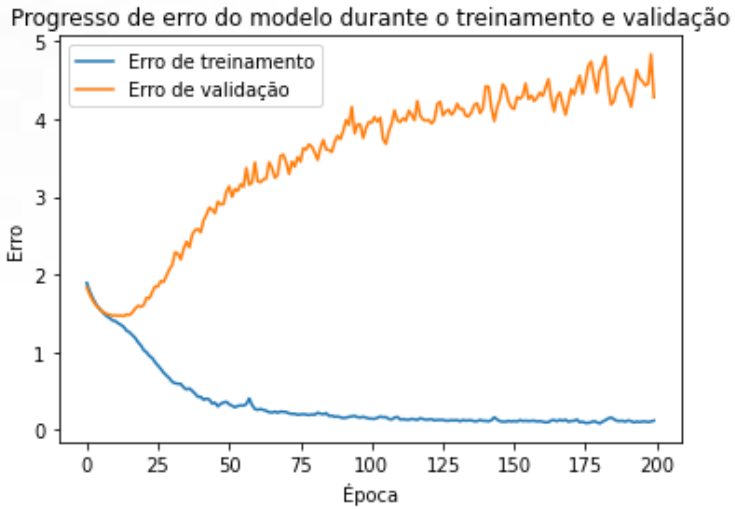
- **Camadas Convolucionais (Conv1D):**
  - conv1d: A primeira camada convolucional com 6432 parâmetros (somatório da quantidade de pesos em cada filtro: (*2 (tamanho do filtro) * 100 + 1) * 32).*
  - conv1d_1: A segunda camada convolucional com 9632 parâmetros. *(filtro de tamanho 3)*
  - conv1d_2: A terceira camada convolucional com 12832 parâmetros. *(filtro de tamanho 4)*
  - conv1d_3: A quarta camada convolucional com 16032 parâmetros. *(filtro de tamanho 5)*
- **Camada de Max Pooling Global (GlobalMaxPooling1D):**
  - global_max_pooling1d: Camada de max pooling global que não possui parâmetros treináveis. (saída igual a 4*32 (qtd de filtros))
- **Camadas Densas:**
  - dense: Primeira camada densa com 8256 parâmetros. *(4 * 32 (qtd de filtros) * 64 (qtd de neurônios na camada) + 64 (bias de cada neurônio))*
  - dense_1: Segunda camada densa com 4160 parâmetros *(64 (qtd de neurônios) * 64 (qtd de neurônios) + 64 (bias de cada neurônio)).*
  - dense_2: Terceira camada densa com 4160 parâmetros. *(64 (qtd de neurônios) * 64 (qtd de neurônios) + 64 (bias de cada neurônio))*
  - dense_3: Quarta camada densa com 260 parâmetros. *(64 (qtd de neurônios) * 4 (um para cada classe) + 64 (bias de cada neurônio))*
- **Camada de Dropout:**
  - dropout: Camada de dropout que não possui parâmetros treináveis.

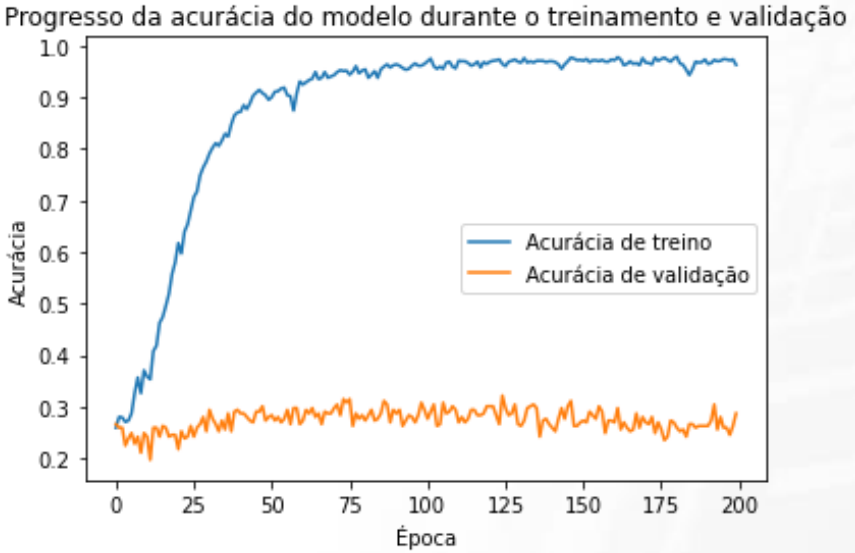
Fases do processo: Treinamento do modelo de Deep Learning

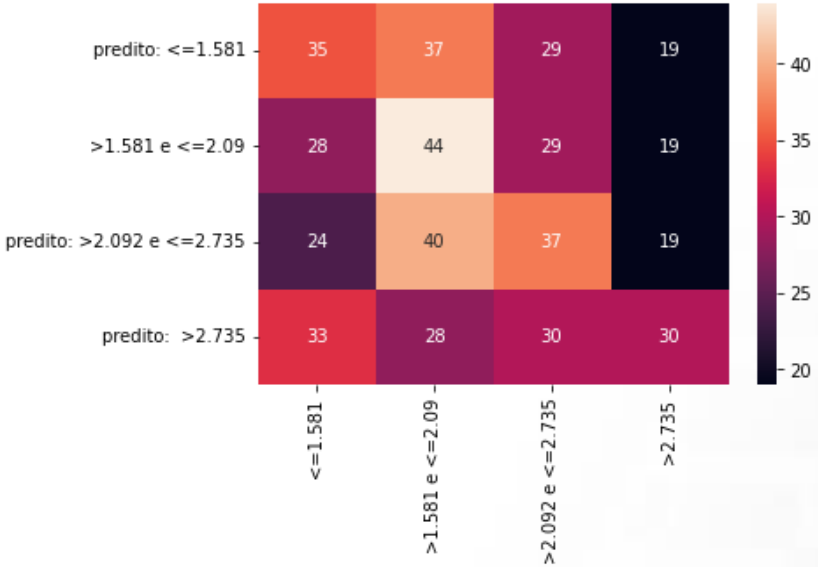
Treinamento foi feito em 200 épocas (iterações), tamanho do batch de 128, utilizando função de perda “sparse_categorical_crossentropy” e o otimizador “adam”;

Fases do processo: Avaliação do modelo

Conclusão: Overffiting. Os dados foram insuficientes. Acurácia de 0.30 nos dados de teste.

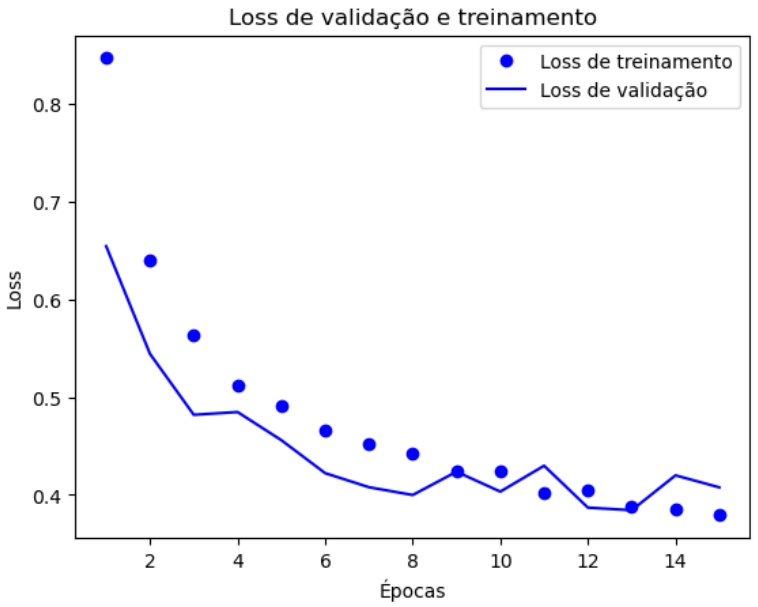


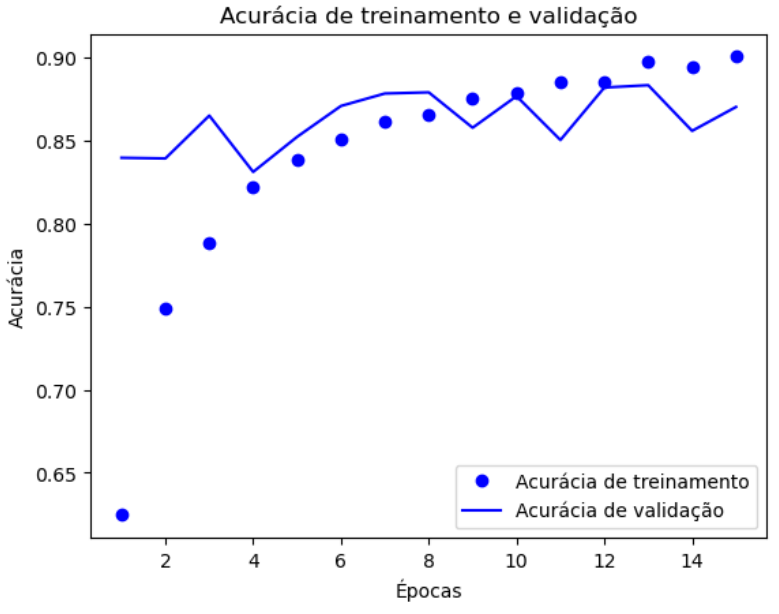




Exemplo de treinamento +ou- bem sucedido

Acurácia de 0.88 nos dados de teste.





Fases do processo: Avaliação do modelo

"A capacidade dos modelos de aprendizagem profunda de aprender features por conta própria depende da disponibilidade de muitos dados de treinamento; se você tiver apenas algumas amostras, o valor da informação em seus features torna-se crítico."

François Chollet – Deep Learning with Python

Aplicações de IA para o Inep

- Estudo dos modelos generativos de código aberto para prospectar aplicações baseadas em IA;
- Treinamento de modelos de word embeddings ou outras representações, como o Transformer, com todo o acervo de itens do Inep (corpus) utilizar no treinamento de modelos de *deep learning* para diversas aplicações;
- Treinamento de modelos de word embeddings ou outras representações com todo acervo de respostas abertas de questionários para utilizar em diversas aplicações;
- Modelo generativo de itens de prova, treinado com o acervo de itens;
- Continuação dos estudos de classificação e regressão de parâmetros;
- Previsão de médias de proficiência do Saeb das próximas avaliações com base nos dados históricos;
- Conversão manuscrito->caractere de folhas de resposta (deep learning para visão computacional, reconhecimento de caracteres);
- Automatização da seleção colaboradores do Inep utilizando ferramentas de IA: reconhecimento de documentação válida, análise de requisitos em currículos lattes, etc;

- https://github.com/johanessevero/classificacao_itens_enem_projeto_final_puc
- https://github.com/johanessevero/modelo_skip_gram_enem