Classification of CPP

Application of a Multilayer Neural Network

Almiro Moreira | Maria Ferreira | David Santos | Ana Carmona | Rui Alves

18 july 2022

Problem Definition

Problem Definition

1-digit CPP classification, Census2011

Multi-class: **10 classes**

Application of a multilayer neural network

What is CPP?

The CPP is the set of all existing occupations in Portugal and their respective functional description, presented together by professional groups.

It is a fundamental tool for statistics on occupations, both in terms of observation, analysis, consolidation of series and statistical technical coordination, and for statistical comparability at European and international level at all these common levels.

Large Group CPP (single digit)

- 0 Armed Forces Professions
- 1 Representatives of the legislature and executive bodies, directors, directors and executive managers
- 2 Specialists in intellectual and scientific activities
- 3 Intermediate level technicians and professions
- 4 Administrative staff
- 5 Personal, security and safety service workers and vendors
- 6 Farmers and skilled workers in agriculture, hunting, fishing and forestry
- 7 Skilled workers from industry, construction and craftsmen
- 8 Plant and machine operators and assembly workers
- 9 unskilled workers

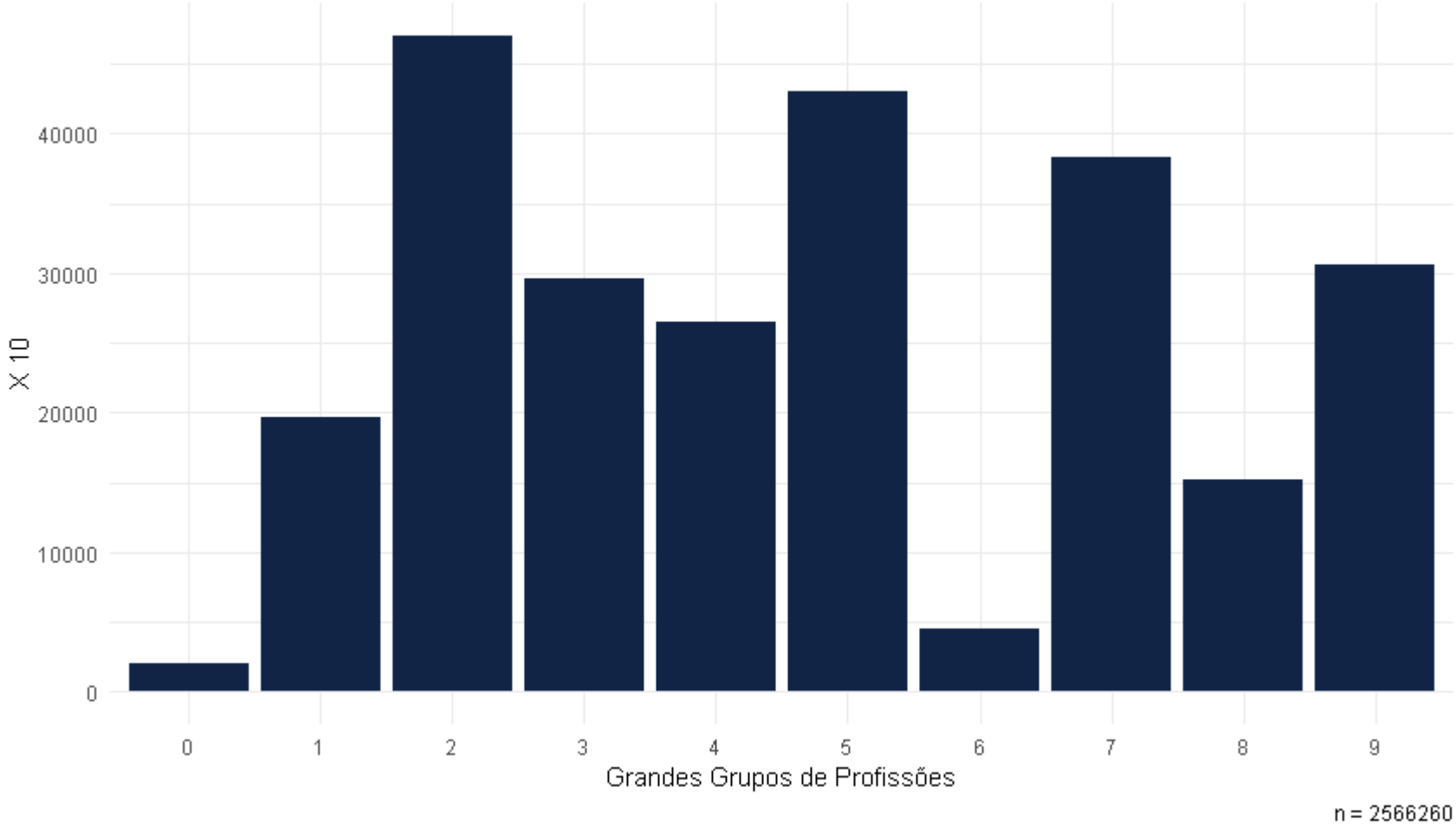
CPP: Coding system and structure

Níveis	Total Classes	Breakdown/digits
Large Group	10	1
Sub-Large Group	43	2
Sub-Group	130	3
Base Group	442	4
Occupation	708	5

CPP: Encoding example: “Physicist”

Código	Designação	Níveis
2	Experts in scientific intellectual activities	Large Group
21	Specialists in the physical, mathematical, engineering and related technical sciences	Sub-Large Group
211	Physicists, Chemists and Related Experts	Sub-Group
2111	Physicists and Astronomers	Base Group
2111.1	Physicist	Occupation

CPP Large Groups: distribution in the Census2011



The other problem…

Data sets
in tutorials

Data sets in
the wild

Pre-Processing

Pre-Processing

4 variables:

`Occupation`, **`company activity`**, **`product of the company`** and **`tasks`**

- lowercase conversion
- codepage enforcement
- removal of accents
- removal of special characters
- removal of specific *stopwords*
- removal of double spaces, spaces at the beginning and at the end

Pre-Processing

Original Data

```
##                               PROFISSAO ACTIV_EMPRESA   PRODS_EMP_COD
## 1   Professora do Ensino Básico-1º ciclo  EBI da Horta educação/ensino
## 2 Tecnica Especializada Telecomunicaçoes   outsourcing     mao de obra
## 3                  Técnico de Informatica     comércio        serviços
## 4                             Investigador       Ensino          Ensino
## 5                                Professor       Ensino          Ensino
##                      TAREFAS CPP
## 1   ensino a ler e escrever   2
## 2        supervisao do GNOC   3
## 3 Administrador de sistemas   3
## 4  Investigação e leccionar   2
## 5                Dar aulas   2
```

Pre-Processing

Data After Pre-Processing

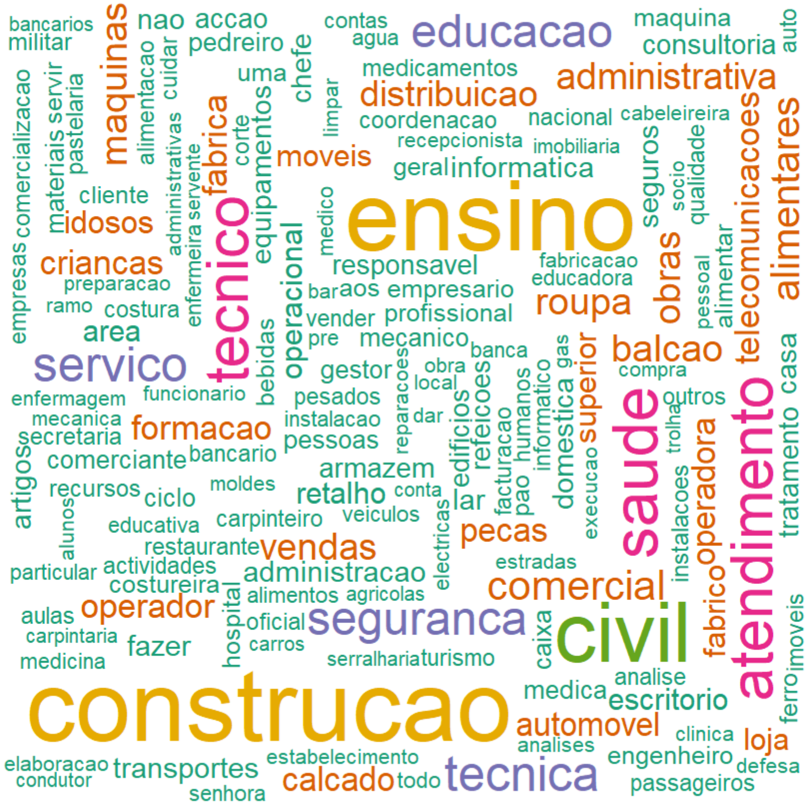
```
##                                PROFISSAO ACTIV_EMPRESA   PRODS_EMP_COD
## 1      professora ensino basico 1o ciclo     ebi horta educacao ensino
## 2 tecnica especializada telecomunicacoes   outsourcing        mao obra
## 3                    tecnico informatica     comercio        servicos
## 4                             investigador       ensino          ensino
## 5                                professor       ensino          ensino
##                   TAREFAS CPP
## 1    ensino ler escrever   2
## 2        supervisao gnoc   3
## 3 administrador sistemas   3
## 4 investigacao leccionar   2
## 5              dar aulas   2
```

Pre-Processing

Concatenation of information from the 4 variables

```
##                                                                    cpp_dsg
## 1 professora ensino basico 1o ciclo ensino ler escrever ebi horta educacao ensino
## 2     tecnica especializada telecomunicacoes supervisao gnoc outsourcing mao obra
## 3                    tecnico informatica administrador sistemas comercio servicos
## 4                              investigador investigacao leccionar ensino ensino
## 5                                          professor dar aulas ensino ensino
##   CPP
## 1   2
## 2   3
## 3   3
## 4   2
## 5   2
```

Word Cloud



bancarios
militar
maquinas
nao
accao
pedreiro
contas
agua
educacao
maquina
consultoria
auto
comercializacao
materiais
servir
pastelaria
alimentacao
cuidar
uma
chefe
medicamentos
distribuicao
administrativa
limpar
coordenacao
nacional
cabeleireira
administrativas
fabrica
corte
equipamentos
moveis
recepcionista
imobiliaria
geral
informatica
seguros
socio
qualidade
telecomunicacoes
alimentares
cliente
idosos
servente
enfermeira
medico
ensino
criancas
preparacao
responsavel
fabricacao
educadora
obras
pessoal
alimentar
empresas
ramo
costura
bar
aos
empresario
vender
profissional
roupa
tecnico
operacional
area
pre
mecanico
banca
gas
balcao
compra
servico
bebidas
gestor
obra
local
enfermagem
funcionario
pesados
reparacoes
edificios
refeicoes
humanos
informatico
domestica
superior
saude
outros
casa
mecanica
formacao
instalacao
dar
facturacao
pao
execucao
trolha
operadora
atendimento
tratamento
secretaria
pessoas
artigos
comerciante
bancario
armazem
recursos
moldes
retalho
conta
lar
ciclo
veiculos
alunos
educativa
carpinteiro
electricas
pecas
instalacoes
restaurante
vendas
estradas
particular
actividades
administracao
costureira
alimentos
agricolas
comercial
aulas
operador
hospital
oficial
seguranca
caixa
civil
fabrico
imoveis
carpintaria
fazer
carros
medicina
serralharia
turismo
analise
construcao
medica
escritorio
ferro
automovel
clinica
loja
analises
elaboracao
transportes
estabelecimento
engenheiro
defesa
condutor
calcado
todo
tecnica
passageiros
senhora

The solution

Input preparation

Training set and Test set

- Stratification by CPP

Training set: 70% (1649803)

Test Set: 30% (721612)

CPP	0	1	2	3	4	5	6	7	8	9
Treino	12953	133976	286348	199270	181177	280126	27463	238689	99184	190617
Teste	5670	58009	125960	86441	78433	122406	12268	105204	43496	83725

Input preparation

Tokenizers

Tokenize the text: convert the text into a integer sequence in which each number corresponds to a word in the dictionary.

```
texts[[1]]
```

[1] “militar logistica defesa soberania nacional defesa nacional”

```
sequences_treino[1]
```

[[1]] [1] 158 210 247 2996 185 247 185

```
paste(collapse=' ',tokenizer$index_word[sequences[[1]]])
```

[1] “militar logistica defesa soberania nacional defesa nacional”

Input preparation

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]	[,9]	[,10]	[,11]	[,12]	[,13]	[,14]	[,15]	[,16]	[,17]	[,18]	[,19]	[,20]
[1,]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
[2,]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
[3,]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
[4,]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
[5,]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
[6,]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	[,21]	[,22]	[,23]	[,24]	[,25]	[,26]	[,27]	[,28]	[,29]	[,30]	[,31]	[,32]	[,33]	[,34]	[,35]	[,36]	[,37]	[,38]	[,39]
[1,]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
[2,]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1409	1514	2698
[3,]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	158	710	590	558	1234
[4,]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
[5,]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
[6,]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	[,40]	[,41]	[,42]	[,43]	[,44]	[,45]	[,46]	[,47]
[1,]	0	158	210	247	2996	185	247	185
[2,]	483	91	91	155	174	16117	91	287
[3,]	710	590	558	19	247	408	9142	397
[4,]	158	1	400	21	18812	3523	247	185
[5,]	0	0	175	710	590	78	247	247
[6,]	0	158	1514	2698	528	338	247	2969

... dim: 1649803 x 47

Input preparation

One hot Encoding

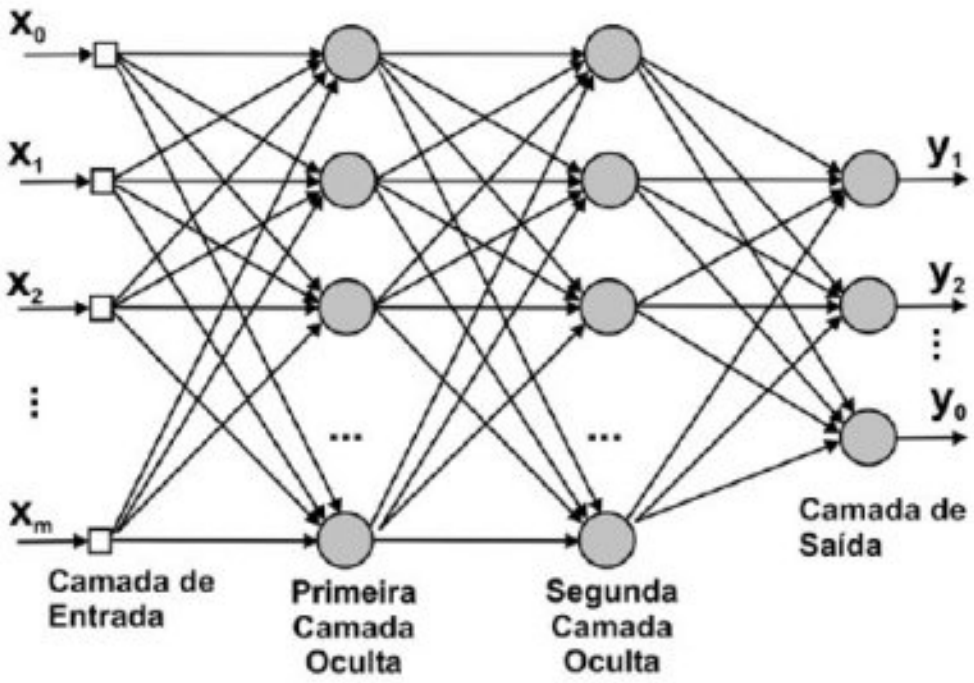
Before starting to build the model, it is necessary to transform the CPP vector, which has a value for each class, into a boolean matrix.

The *package* ***keras*** has a function for this (***to_categorical()***).

```
labels <- to_categorical(labels, num_classes = 10)
```

	0	1	2	3	4	5	6	7	8	9
1	1	0	0	0	0	0	0	0	0	0
2	0	1	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	1	0	0	0
...	...	...	...	...	...	...	...	...	...	...
1649803	0	0	0	0	0	0	0	0	1	0

The model


x0
x1
x2
xm
y1
y2
y0
Camada de Entrada
Primeira Camada Oculta
Segunda Camada Oculta
Camada de Saída

The model

```
embedding_dim <- 64
maxlen <- 47
max_words <- 30000

model <- keras_model_sequential() %>%
         layer_embedding(input_dim = max_words,
                         output_dim = embedding_dim,
                         input_length = maxlen) %>%
         layer_flatten() %>%
         layer_dense(units = 512, activation = 'relu') %>%
         layer_dropout(rate = 0.5) %>%
         layer_dense(units = 256, activation = 'relu') %>%
         layer_dropout(rate = 0.5) %>%
         layer_dense(units = 10, activation = 'softmax')
```

Model Summary

```
Model: "sequential_3"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_3 (Embedding)     (None, 47, 64)            1920000

 flatten_3 (Flatten)         (None, 3008)              0

 dense_11 (Dense)            (None, 512)               1540608

 dropout_7 (Dropout)         (None, 512)               0

 dense_10 (Dense)            (None, 256)               131328

 dropout_6 (Dropout)         (None, 256)               0

 dense_9 (Dense)             (None, 10)                2570

=================================================================
Total params: 3,594,506
Trainable params: 3,594,506
Non-trainable params: 0
```

Compile and tweak the model

After configuring the model's architecture, it is necessary to compile and adjust it to improve performance.

For compilation we use the `compile()` function and parameterize *loss* and *optimizer*.

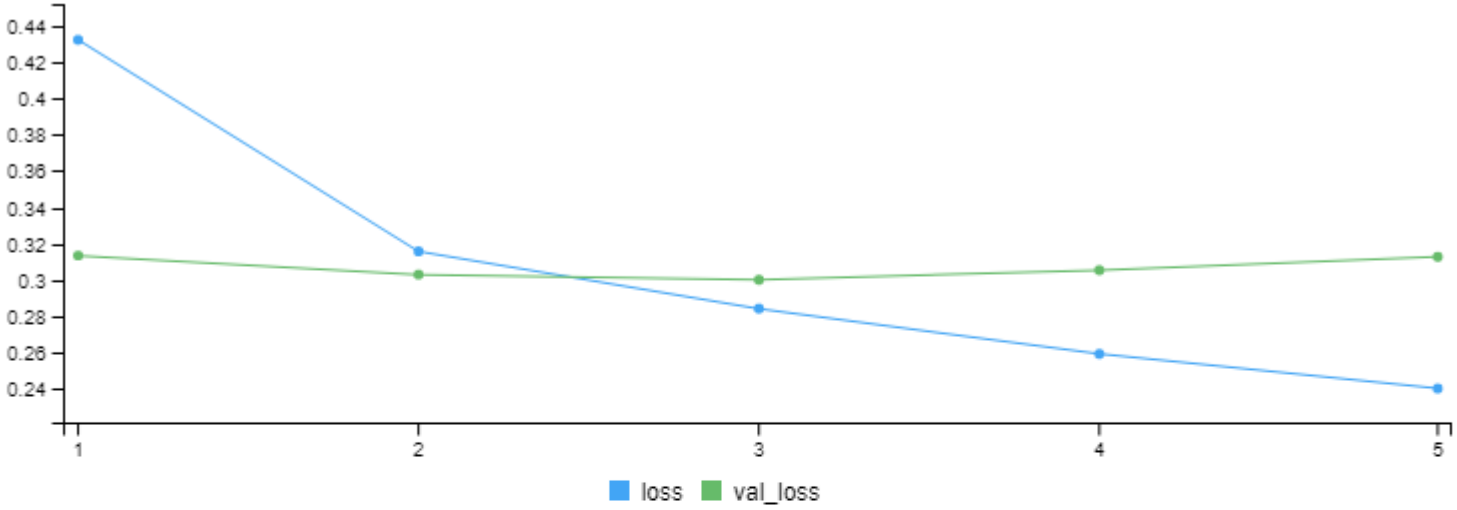
Compile the template

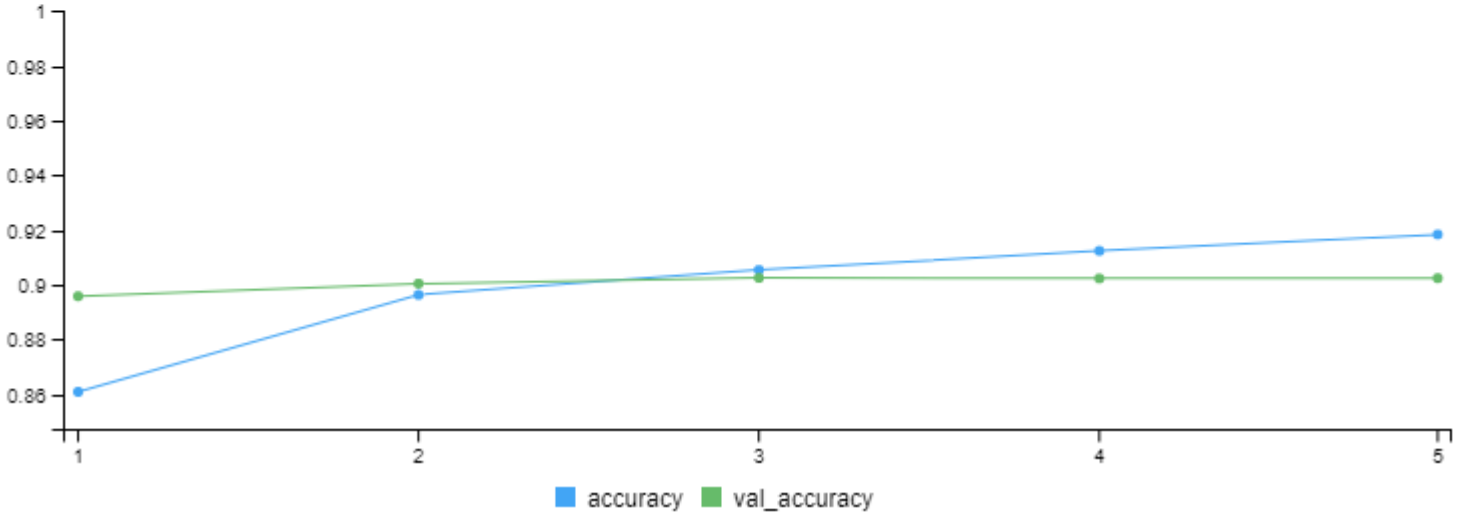
```
model   %>% compile(optimizer = 'adam',
                    loss='categorical_crossentropy',
                    metrics = 'accuracy')
```

Adjust the model

```
history <- model %>%
           fit(data[ix,],
           labels[ix,],
           epochs = 5,
           batch_size = 256,
           validation_data = list(data[-ix,],
           labels[-ix,]))
```

Training history





Predict Value

predict from classes to test data

```
predictions <- model %>%
               predict(data_test) %>%
               k_argmax() %>%
               as.double()
```

```
predictions
     0      1      2      3      4      5      6      7      8      9
  5898  59610 128204  83734  77919 121302  11950 105243  43785  83967
```

Model Evaluation

```
results <- model %>%
            evaluate(data_test, labels_test)
```

results

```
  loss           accuracy

  0.3085286      0.9043350
```

Confusion Matrix (%)

Reais \ Previstos	0	1	2	3	4	5	6	7	8	9
0	98,3	0,2	0,3	0,3	0,3	0,3	0,0	0,1	0,1	0,1
1	0,0	85,5	2,5	2,4	0,6	6,8	0,2	1,8	0,1	0,1
2	0,1	1,2	96,0	1,6	0,5	0,4	0,0	0,2	0,0	0,1
3	0,1	4,4	5,9	80,5	3,3	2,7	0,1	1,9	0,3	0,8
4	0,1	0,5	1,6	3,5	90,1	2,1	0,0	0,8	0,2	1,2
5	0,1	3,1	0,8	1,4	1,0	91,4	0,0	0,4	0,1	1,7
6	0,0	1,5	0,5	0,7	0,2	0,6	92,8	0,5	0,4	2,7
7	0,0	0,5	0,3	1,7	0,2	0,4	0,0	94,5	1,0	1,4
8	0,1	0,3	0,1	1,1	0,4	0,4	0,2	5,3	88,4	3,8
9	0,0	0,2	0,3	0,8	1,2	3,4	0,3	2,5	1,8	89,4

Packages used

Packages used

This presentation was made in Rmarkdown > Presentation > HTML (ioslides)

janitor Sam Firke (2021). janitor: Simple Tools for Examining and Cleaning Dirty Data. R package. version 2.1.0. https://CRAN.R-project.org/package=janitor

keras Kuhn et al., (2020). Tidymodels: a collection of packages for modeling and machine learning using tidyverse principles. https://www.tidymodels.org

qdap Rinker, T. W. (2020). qdap: Quantitative Discourse Analysis Package. 2.4.2. Buffalo, New York. https://github.com/trinker/qdap

tidyverse Wickham et al., (2019). Welcome to the tidyverse. Journal of Open Source Software, 4(43), 1686, https://doi.org/10.21105/joss.01686

tidymodels Kuhn et al., (2020). Tidymodels: a collection of packages for modeling and machine learning using tidyverse principles. https://www.tidymodels.org

Thank you for your attention!