

DSCI353-353m-453 & E1453-E2453: Class 10b Fundamentals of ML with NN

Profs: R. H. French, L. S. Bruckman, P. Leu, K. Davis, S. Cirlos

TAs: W. Oltjen, K. Hernandez, M. Li, M. Li, D. Colvin

21 February, 2023

Contents

6.1.1.1	Class Readings, Assignments, Textbooks Syllabus Topics	2
6.1.1.1.1	Reading, Lab Exercises, SemProjects	2
6.1.1.1.2	Textbooks	2
6.1.1.1.3	Syllabus	2
6.1.1.2	Introduction to Machine Learning and Deep Learning with Neural Networks	4
6.1.1.3	Fundamentals of machine learning	5
6.1.1.3.1	Generalization: The goal of machine learning	5
6.1.1.3.2	Adding white-noise channels or all-zeros channels to MNIST	5
6.1.1.3.3	Training the same model on MNIST data with noise channels or all-zero channels	5
6.1.1.3.4	Plotting a validation accuracy comparison	6
6.1.1.4	The nature of generalization in deep learning	7
6.1.1.4.1	Fitting a MNIST model with randomly shuffled labels . . .	7
6.1.1.5	The manifold hypothesis	8
6.1.1.6	Training a MNIST model with an incorrectly high learning rate . .	9
6.1.1.6.1	The same model with a more appropriate learning rate . .	9
6.1.1.7	Leveraging better architecture priors	10
6.1.1.7.1	Increasing model capacity	10
6.1.1.8	Improving generalization	11
6.1.1.8.1	Adding weight regularization	12
6.1.1.8.2	Adding weight regulariza	13
6.1.1.8.3	Adding dropout	13
6.1.1.9	Links	14

```
knitr::opts_chunk$set(  
  cache = FALSE, # if TRUE knitr will cache results to reuse in future knits  
  fig.width = 6, # the width for plots created by code chunk  
  fig.height = 4, # the height for plots created by code chunk  
  fig.align = 'center', # how to align graphics. 'left', 'right', 'center'  
  dpi = 300,  
  dev = 'png', # Makes each fig a png, and avoids plotting every data point  
  # eval = FALSE, # if FALSE, then the R code chunks are not evaluated  
  # results = 'asis', # knitr passes through results without reformatting  
  echo = TRUE, # if FALSE knitr won't display code in chunk above it's results  
  message = TRUE, # if FALSE knitr won't display messages generated by code  
  strip.white = TRUE, # if FALSE knitr won't remove white spaces at beg or end of code chunk  
  warning = TRUE, # if FALSE knitr won't display warning messages in the doc
```

```
error = TRUE) # report errors
# options(tinytex.verbose = TRUE)
```

6.1.1.1 Class Readings, Assignments, Textbooks Syllabus Topics

6.1.1.1.1 Reading, Lab Exercises, SemProjects

- Readings:
 - For today: ISLR7
 - For next class: ISLR10 (R4DS22-25)
- Laboratory Exercises:
 - LE3 : is due Thursday
- Office Hours: (Class Canvas Calendar for Zoom Link)
 - **Mondays @ 4:00 PM to 5:00 PM**
 - Wednesdays @ 4:00 PM to 5:00 PM
 - Office Hours are on Zoom, and recorded
- Semester Projects
 - DSCI 453 Students Biweekly Updates Due
 - DSCI 453 Students
 - * **Next Report Out #2 is Due This Friday March 18th**
 - SemProj Office Hours, with Prof. Bruckman
 - * Tuesdays at 9am to 10am.
 - * Zoom Invites are in Case Canvas
 - All DSCI 353/353M/453, E1453/2453 Students:
 - * **Peer Grading of Report Out 2 is Due March 25th**
 - Exams
 - * Final: Thursday March 28th, 2022, 12:00PM - 3:00PM, Nord 356 or remote

6.1.1.1.2 Textbooks

- Text Books for DSCI353/353M/453
 - [R4DS: Wickham: R for Data Science](#)
 - [ISLR: Intro to Statistical Learning with R, 2nd Ed.](#)
 - DLwR: Deep Learning with R, Chollet, Allaire,
 - [DLGB: Deep Learning, Goodfellow, Bengio, Courville](#)
- Magazine Articles about Deep Learning
 - DL1 to DL12 are “Deep Learning” articles in 3-readings/2-articles/

6.1.1.1.3 Syllabus

Day:Date	Foundation	Practicum	Readings(optional)	Due(optional)
w01a:Tu:1/17/23	Markov Cluster	R, Rstudio IDE, Git		(LE0)
w01b:Th:1/19/23	Stat. Learning, Approach	Bash, Git, Class Repo	ISLR1,2 (R4DS-1-3)	
w02a:Tu:1/24/23	Lin. Regr. Bias-Var.	SemProjs; Regr. Ovrw	ISLR3,(R4DS-4-6)	(LE0:Due) LE1 453 Update 1
w02b:Th:1/26/23	Train/Test, Bias vs. Vari.	Tidyverse Review	DL01 DL02 (R4DS-7,8)	
w02Pr:Fr:1/27/23	ADD DROP	DEADLINE		
w03a:Tu:1/31/23	Logistic Regr. Classif	Pred. Analytics, Regr.	DL03,ISLR4	LE1:Due , LE2 LE1:Due
w03b:Th:2/2/23	LDA/QDA	ggPlot2, Code Expect.	DL04, DL05	
w03Sa:2/4/23				
w04a:Tu:2/7/23	Resample Cross-Valid.	ggplot	ISLR5	453 Update 2 LE2:Due , LE3 453 Rep. Out 1
w04b:Th:2/9/23	DL, ML Overview	Multilevel Mod.	ISLR6 (R4DS9-16)	
w04Pr:Fr:2/10/23				
w05a:Tu:2/14/23	Bootstrap	Bootstrap Mixed Effects	DL2R1, DL06,07	453 Rep. Out 1
w05b:Th:2/16/23	Subset Selec., Shrink.	Mixed Effects	DLwR2	
w05Pr:Fr:2/17/23				
w06a:Tu:2/21/23	Mod. Selec.	ML with NNs	DLwR3	LE3:Due , LE4 453 Update 3
w06b:Th:2/23/23	Beyond Linear Modls	Feature Select., Caret	ISLR7 (R4DS22-25)	
w06Pr:Fr:2/24/23				
w07a:Tu:2/28/23	Dec. Trees, Rand. Forest.	Tidy Modeling	ISLR8, DL08,09	
w07b:Th:3/2/23	MidTerm Review, SVM	SVM, SVR, ROC	ISLR9 (R4DS26-30)	Peer Review 1
w08a:Tu:3/7/23	R-Keras/TensorFlow2		ISLR10	
w08b:Th:3/9/23	MIDTERM EXAM		DL10,11	LE4:Due LE5
w08Pr:Fr:3/10/23				453 Update 4
Tu:3/14/23	SPRING	BREAK	ISLR10	
Th:3/16/23	SPRING	BREAK	DL12,13	
w09a:Tu:3/21/23	Deep Learning	TF2 Keras Intro	Pocket Perceptron	ISLR10, DLR3
w09b:Th:3/23/23	Computer Vision, CNN	CNN w/TF2, Overfit	DLR4	
w09Pr:Fr:3/24/23				453 Rep. Out 2
w10a:Tu:3/28/23	Deep Learn Intro	NN Types	DLR5	
w10b:Th:3/30/23	DL CNN,RNN ImageNet	NN Types, CNN w/TF2	Hinton ImageNet	
w10Pr:Fr:3/31/23				453 Upd.5 & PrRev 2 LE5:Due LE6
Sa:4/1/23				
w11a:Tu:4/4/23	Fitting NNs	AUC,Pre,Recall Fruit		LE6:Due LE7
w11b:Th:4/6/23	NLP, Graphs & ML		LeCun DL Rev. 2015	
w12a:Tu:4/11/23	Graphs & ML	NLP with sequences	DLR6	
w12b:Th:4/13/23	NLP w attention	Graph Repr Proc Wrk-flw		
w13a:Tu:4/18/23	DL Frameworks	Explaining DL w Lime		
w13b:Th:4/20/23	Linux Distros XGBoost	Explain Preds	Deep Dream	
w13Pr:Fr:4/21/23				453 Rep. Out 3 Due
w14a:Tu:4/25/23	Transformers			LE7:Due Peer Rev 3 Due
w14b:Th:4/27/23	Final Exam Review	Torch NN & DeepLearn		
w14Pr:Fr:4/28/23	FINAL EXAM	Th. 5/4/23, 12-3pm	Nord 356 & Zoom	
	453 Final PDF Report	Fr. 4/29, 11:59pm		

Figure 1: DSCI351-351M-451 Syllabus

6.1.1.2 Introduction to Machine Learning and Deep Learning with Neural Networks

- This is a companion notebook for the book
 - [Deep Learning with R, Second Edition](#).

For readability,

- it only contains runnable code blocks and section titles,
- and omits everything else in the book: text paragraphs, figures, and pseudocode.

If you want to be able to follow what's going on,

- I recommend reading the notebook side by side with your copy of the book

This notebook was generated for TensorFlow 2.7.

This is using the [MNIST dataset](#)



Figure 2: MNIST digist

MNIST images through a Neural Network Model layers

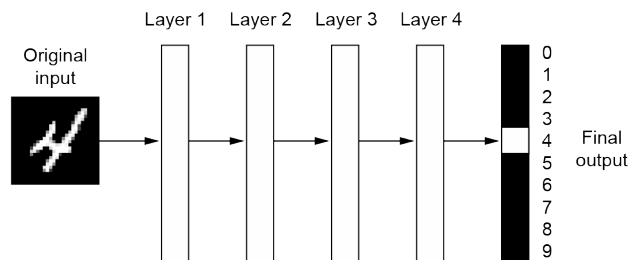


Figure 3: MNIST images through a Neural Network Model layers

MNIST and how the layers learn representations

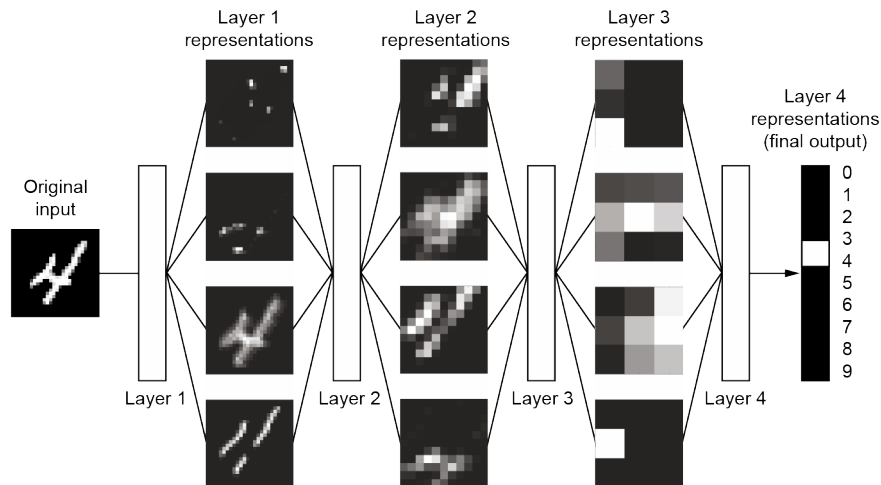


Figure 4: MNIST and how the layers learn representations.

6.1.1.3 Fundamentals of machine learning

6.1.1.3.1 Generalization: The goal of machine learning

- Underfitting and overfitting
- Noisy training data
- Ambiguous features
- Rare features and spurious correlations

```
library(keras)

mnist <- dataset_mnist()
train_labels <- mnist$train$y
train_images <- mnist$train$x %>%
  array_reshape(c(60000, 28 * 28))
train_images <- train_images / 255

noise_channels <- array(runif(nrow(train_images) * 784),
  dim = c(nrow(train_images), 784))
train_images_with_noise_channels <-
  cbind(train_images, noise_channels)

zeros_channels <- array(0, dim = c(nrow(train_images), 784))
train_images_with_zeros_channels <-
  cbind(train_images, zeros_channels)
```

6.1.1.3.2 Adding white-noise channels or all-zeros channels to MNIST

```
library(keras)

get_model <- function() {
  model <- keras_model_sequential() %>%
    layer_dense(512, activation = "relu") %>%
    layer_dense(10, activation = "softmax")

  model %>% compile(optimizer = "rmsprop",
    loss = "sparse_categorical_crossentropy",
    metrics = "accuracy")

  model
}
```

```
model <- get_model()
history_noise <- model %>% fit(
  train_images_with_noise_channels,
  train_labels,
  epochs = 10,
  batch_size = 128,
  validation_split = 0.2
)
```

6.1.1.3.3 Training the same model on MNIST data with noise channels or all-zero channels

```
## Error in eval(expr, envir, enclos): tensorflow.python.framework.errors_impl.InternalError: Failed copy
model <- get_model()
```

```
## Error in eval(expr, envir, enclos): tensorflow.python.framework.errors_impl.ResourceExhaustedError:
history_zeros <- model %>% fit(
  train_images_with_zeros_channels,
  train_labels,
  epochs = 10,
  batch_size = 128,
  validation_split = 0.2
)
```

```
## Error in eval(expr, envir, enclos): tensorflow.python.framework.errors_impl.InternalError: Failed copy
```

6.1.1.3.4 Plotting a validation accuracy comparison TODO: plot() should always draw to the graphics device,

- currently when method==ggplot it only returns a plot object
 - which should be the work of autoplot()
- data.frame(
 - “Validation accuracy with noise_channels” = history_noisemetrics\$val_accuracy,
 - “Validation accuracy with zeros channels” = history_zerosmetrics\$val_accuracy
 - %>% plot(type = ‘l’)

```
library(patchwork)

plot(
  NULL,
  NULL,
  main = "Effect of Noise Channels on Validation Accuracy",
  xlab = "Epochs",
  xlim = c(1, history_noise$params$epochs),
  ylab = "Validation Accuracy",
  ylim = c(0.9, 1)
)
```

```
## Error in plot.default(NULL, NULL, main = "Effect of Noise Channels on Validation Accuracy", : object
lines(history_zeros$metrics$val_accuracy,
  lty = 1,
  type = 'o')
```

```
## Error in lines(history_zeros$metrics$val_accuracy, lty = 1, type = "o"): object 'history_zeros' not
lines(history_noise$metrics$val_accuracy,
  lty = 2,
  type = 'o')
```

```
## Error in lines(history_noise$metrics$val_accuracy, lty = 2, type = "o"): object 'history_noise' not
legend(
  "bottomright",
  legend = c(
    "Validation accuracy with zeros channels",
    "Validation accuracy with noise channels"
  ),
```

```
lty = 1:2
)
```

```
## Error in (function (s, units = "user", cex = NULL, font = NULL, vfont = NULL, : plot.new has not been
```

6.1.1.4 The nature of generalization in deep learning

```
c(c(train_images, train_labels), .) %<-% dataset_mnist()
train_images <- train_images %>%
  array_reshape(c(60000, 28 * 28)) %>%
  `/(` (255) #Q: do we teach this?

random_train_labels <- sample(train_labels)

model <- keras_model_sequential() %>%
  layer_dense(512, activation = "relu") %>%
  layer_dense(10, activation = "softmax")

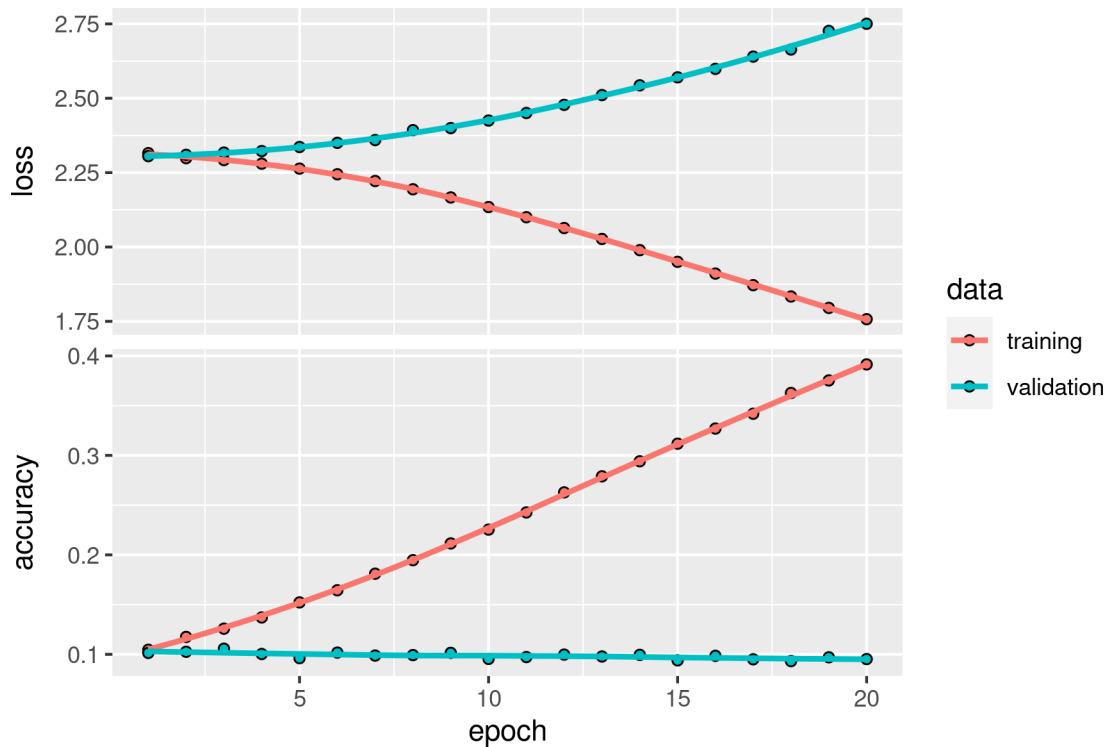
summary(model)
```

6.1.1.4.1 Fitting a MNIST model with randomly shuffled labels

```
## Model: <no summary available, model was not built>
```

```
model %>% compile(optimizer = "rmsprop",
                  loss = "sparse_categorical_crossentropy",
                  metrics = "accuracy")
```

```
history <- model %>% fit(
  train_images,
  random_train_labels,
  epochs = 20, # best use 100 epochs
  batch_size = 128,
  validation_split = 0.2
)
plot(history)
```



6.1.1.5 The manifold hypothesis

- Interpolation as a source of generalization
- Why deep learning works
- Training data is paramount
- Evaluating machine-learning models
- Training, validation, and test sets
- Simple hold-out validation
- K-fold validation
- Iterated K-fold validation with shuffling
- Beating a common-sense baseline
- Things to keep in mind about model evaluation
- Improving model fit
- Tuning key gradient descent parameters

```
c(c(train_images, train_labels), .) %<-% dataset_mnist()
train_images <- train_images %>%
  array_reshape(c(60000, 28 * 28)) %>% `/\`(255)

model <- keras_model_sequential(list(
  layer_dense(units = 512, activation = "relu"),
  layer_dense(units = 10, activation = "softmax")
))
model %>% compile(optimizer = optimizer_rmsprop(1.),
  loss = "sparse_categorical_crossentropy",
  metrics = "accuracy")

model %>% fit(
```



```

train_images,
train_labels,
epochs = 10,
batch_size = 128,
validation_split = 0.2
)

```

6.1.1.6 Training a MNIST model with an incorrectly high learning rate

```

## Error in eval(expr, envir, enclos): tensorflow.python.framework.errors_impl.ResourceExhaustedError: (
## <... omitted ...>st-packages/keras/optimizers/optimizer_experimental/optimizer.py", line 526, in min
##     grads_and_vars = self.compute_gradients(loss, var_list, tape)
##     File "/usr/local/lib/python3.8/dist-packages/keras/optimizers/optimizer_experimental/optimizer.py
##     grads = tape.gradient(loss, var_list)
## Node: 'gradient_tape/sequential_3/dense_6/MatMul/MatMul'
## OOM when allocating tensor with shape[784,512] and type float on /job:localhost/replica:0/task:0/dev
## [[{{node gradient_tape/sequential_3/dense_6/MatMul/MatMul}}]]
## Hint: If you want to see a list of allocated tensors when OOM happens, add report_tensor_allocations.
## [Op:__inference_train_function_39520]
## See `reticulate::py_last_error()` for details

```

```
summary(model)
```

```

## Model: "sequential_3"
## -----
## Layer (type)                Output Shape          Param #
## -----
## dense_6 (Dense)             (128, 512)            401920
## dense_7 (Dense)             (128, 10)              5130
## -----
## Total params: 407,050
## Trainable params: 407,050
## Non-trainable params: 0
## -----

```

```

model <- keras_model_sequential(list(
  layer_dense(units = 512, activation = "relu"),
  layer_dense(units = 10, activation = "softmax")
))
model %>% compile(optimizer = optimizer_rmsprop(1e-2),
  loss = "sparse_categorical_crossentropy",
  metrics = "accuracy")
model %>% fit(
  train_images,
  train_labels,
  epochs = 10,
  batch_size = 128,
  validation_split = 0.2
)

```

6.1.1.6.1 The same model with a more appropriate learning rate

```
## Error in eval(expr, envir, enclos): tensorflow.python.framework.errors_impl.InternalError: Failed copy
```

```
summary(model)
```

```
## Model: <no summary available, model was not built>
```

6.1.1.7 Leveraging better architecture priors

6.1.1.7.1 Increasing model capacity A simple logistic regression on MNIST

```
model <- keras_model_sequential() %>%  
  layer_dense(10, activation = "softmax")  
model %>% compile(optimizer = "rmsprop",  
                  loss = "sparse_categorical_crossentropy",  
                  metrics = "accuracy")  
history_small_model <- model %>% fit(  
  train_images,  
  train_labels,  
  epochs = 20,  
  batch_size = 128,  
  validation_split = 0.2  
)
```

```
## Error in eval(expr, envir, enclos): tensorflow.python.framework.errors_impl.InternalError: Failed copy  
summary(model)
```

```
## Model: <no summary available, model was not built>
```

```
dplyr::as_tibble(history_small_model) %>%  
  dplyr::filter(metric == "loss", data == "validation") %>%  
  plot(  
    value ~ epoch,  
    data = .,  
    type = "o",  
    main = "Effect of Insufficient Model Capacity on Validation Loss",  
    xlab = "Epochs",  
    ylab = "Validation Loss"  
  )
```

```
## Error in dplyr::as_tibble(history_small_model): object 'history_small_model' not found
```

```
# w/o going through data.frame
```

```
plot(  
  history_small_model$metrics$val_loss,  
  type = 'o',  
  main = "Effect of Insufficient Model Capacity on Validation Loss",  
  xlab = "Epochs",  
  ylab = "Validation Loss"  
)
```

```
## Error in plot(history_small_model$metrics$val_loss, type = "o", main = "Effect of Insufficient Model
```

```
model <- keras_model_sequential() %>%  
  layer_dense(96, activation = "relu") %>%  
  layer_dense(96, activation = "relu") %>%  
  layer_dense(10, activation = "softmax")  
  
model %>% compile(optimizer = "rmsprop",
```

```

        loss = "sparse_categorical_crossentropy",
        metrics = "accuracy")
history_large_model = model %>% fit(
  train_images,
  train_labels,
  epochs = 20,
  batch_size = 128,
  validation_split = 0.2
)

```

```

## Error in eval(expr, envir, enclos): tensorflow.python.framework.errors_impl.InternalError: Failed copy
summary(model)

```

```

## Model: <no summary available, model was not built>

```

6.1.1.8 Improving generalization

- Dataset curation
- Feature engineering
- Using early stopping
- Regularizing your model
- Reducing the network's size

Original model

```

c(c(train_data, train_labels), .) %<-% dataset_imdb(num_words = 10000)

vectorize_sequences <- function(sequences, dimension = 10000) {
  results <- matrix(0, nrow = length(sequences), ncol = dimension)
  # results[cbind(seq_along(sequences), sequences)] <- 1
  for (i in seq_along(sequences))
    results[i, sequences[[i]]] <- 1.
  results
}
train_data <- vectorize_sequences(train_data)

model <- keras_model_sequential() %>%
  layer_dense(16, activation = "relu") %>%
  layer_dense(16, activation = "relu") %>%
  layer_dense(1, activation = "sigmoid")

model %>% compile(optimizer = "rmsprop",
  loss = "binary_crossentropy",
  metrics = "accuracy")

history_original <- model %>% fit(
  train_data,
  train_labels,
  epochs = 20,
  batch_size = 512,
  validation_split = 0.4
)

```

```

## Error in eval(expr, envir, enclos): tensorflow.python.framework.errors_impl.InternalError: Failed copy

```

```
plot(history_original)
```

```
## Error in plot(history_original): object 'history_original' not found
```

```
summary(model)
```

```
## Model: <no summary available, model was not built>
```

Version of the model with lower capacity

```
model = keras_model_sequential() %>%  
  layer_dense(4, activation = "relu") %>%  
  layer_dense(4, activation = "relu") %>%  
  layer_dense(1, activation = "sigmoid")  
  
model %>% compile(optimizer = "rmsprop",  
                  loss = "binary_crossentropy",  
                  metrics = "accuracy")  
  
history_smaller_model <- model %>% fit(  
  train_data,  
  train_labels,  
  epochs = 20,  
  batch_size = 512,  
  validation_split = 0.4  
)
```

```
## Error in eval(expr, envir, enclos): tensorflow.python.framework.errors_impl.InternalError: Failed copy
```

```
plot(history_smaller_model)
```

```
## Error in plot(history_smaller_model): object 'history_smaller_model' not found
```

Version of the model with higher capacity

```
model = keras_model_sequential() %>%  
  layer_dense(512, activation = "relu") %>%  
  layer_dense(512, activation = "relu") %>%  
  layer_dense(1, activation = "sigmoid")  
  
model %>% compile(optimizer = "rmsprop",  
                  loss = "binary_crossentropy",  
                  metrics = "accuracy")  
  
history_larger_model <- model %>% fit(  
  train_data,  
  train_labels,  
  epochs = 20,  
  batch_size = 512,  
  validation_split = 0.4  
)
```

```
## Error in eval(expr, envir, enclos): tensorflow.python.framework.errors_impl.InternalError: Failed copy
```

```
plot(history_larger_model)
```

```
## Error in plot(history_larger_model): object 'history_larger_model' not found
```

6.1.1.8.1 Adding weight regularization Adding L2 weight regularization to the model

```

model <- keras_model_sequential() %>%
  layer_dense(16,
    kernel_regularizer = regularizer_l2(0.002),
    activation = "relu") %>%
  layer_dense(16,
    kernel_regularizer = regularizer_l2(0.002),
    activation = "relu") %>%
  layer_dense(1, activation = "sigmoid")

model %>% compile(optimizer = "rmsprop",
  loss = "binary_crossentropy",
  metrics = "accuracy")
history_l2_reg <- model %>% fit(
  train_data,
  train_labels,
  epochs = 20,
  batch_size = 512,
  validation_split = 0.4
)

```

```

## Error in eval(expr, envir, enclos): tensorflow.python.framework.errors_impl.InternalError: Failed copy
plot(history_l2_reg)

```

```

## Error in plot(history_l2_reg): object 'history_l2_reg' not found

```

6.1.1.8.2 Adding weight regularization Different weight regularizers available in Keras

```

regularizer_l1(0.001)

```

```

## <keras.regularizers.L1 object at 0x7fea8ae2c5b0>

```

```

regularizer_l1_l2(l1 = 0.001, l2 = 0.001)

```

```

## <keras.regularizers.L1L2 object at 0x7fea8ae2c5e0>

```

6.1.1.8.3 Adding dropout Adding dropout to the IMDB model

```

model <- keras_model_sequential() %>%
  layer_dense(16, activation = "relu") %>%
  layer_dropout(0.5) %>%
  layer_dense(16, activation = "relu") %>%
  layer_dropout(0.5) %>%
  layer_dense(1, activation = "sigmoid")

model %>% compile(optimizer = "rmsprop",
  loss = "binary_crossentropy",
  metrics = "accuracy")
history_dropout <- model %>% fit(
  train_data,
  train_labels,
  epochs = 20,
  batch_size = 512,
  validation_split = 0.4
)

```

```

## Error in eval(expr, envir, enclos): tensorflow.python.framework.errors_impl.InternalError: Failed copy

```

```
plot(history_dropout)
```

```
## Error in plot(history_dropout): object 'history_dropout' not found
```

6.1.1.9 Links

- Chollet, Allaire, Deep Learning with R, 2nd Edition