

# Go Wild

Gerhard Viljoen

21/08/2019

```
rm(list=ls())

require(jsonlite)
require(readtext)

#PDG codes
# pdg.elec <- c(11,-11)
# pdg.pion <- c(-211,211)

files <- list.files(path="C:/Users/gerhard/Documents/msc-thesis-data/processed/000265309", pattern="*.json")

j <- fromJSON(files[1])

for(i in 2:length(files)){

  f <- fromJSON(files[i])
  j <- c(j,f)
}

l <- length(j)

runNumber <- rep(265309,l)

files <- list.files(path="C:/Users/gerhard/Documents/msc-thesis-data/processed/000265332", pattern="*.json")

for(i in 1:length(files)){

  f <- fromJSON(files[i])
  j <- c(j,f)
}

prev_l <- l
l <- length(j)
l <- l-prev_l

runNumber <- c(runNumber,rep(265332,l))

files <- list.files(path="C:/Users/gerhard/Documents/msc-thesis-data/processed/000265334", pattern="*.json")

for(i in 1:length(files)){

  f <- fromJSON(files[i])
  j <- c(j,f)
}

prev_l <- l
l <- length(j)
```

```

l <- l-prev_l

runNumber <- c(runNumber,rep(265334,1))

files <- list.files(path="C:/Users/gerhard/Documents/msc-thesis-data/processed/000265335", pattern="*.js")

for(i in 1:length(files)){

  f <- fromJSON(files[i])
  j <- c(j,f)
}

prev_l <- l
l <- length(j)
l <- l-prev_l

runNumber <- c(runNumber,rep(265335,1))

files <- list.files(path="C:/Users/gerhard/Documents/msc-thesis-data/processed/000265336", pattern="*.js")

for(i in 1:length(files)){

  f <- fromJSON(files[i])
  j <- c(j,f)
}

prev_l <- l
l <- length(j)
l <- l-prev_l

runNumber <- c(runNumber,rep(265336,1))

files <- list.files(path="C:/Users/gerhard/Documents/msc-thesis-data/processed/000265338", pattern="*.js")

for(i in 1:length(files)){

  f <- fromJSON(files[i])
  j <- c(j,f)
}

prev_l <- l
l <- length(j)
l <- l-prev_l

runNumber <- c(runNumber,rep(265338,1))

files <- list.files(path="C:/Users/gerhard/Documents/msc-thesis-data/processed/000265339", pattern="*.js")

for(i in 1:length(files)){

  f <- fromJSON(files[i])
  j <- c(j,f)
}

```

```

prev_l <- 1
l <- length(j)
l <- l-prev_l

runNumber <- c(runNumber,rep(265339,1))

files <- list.files(path="C:/Users/gerhard/Documents/msc-thesis-data/processed/000265342", pattern="*.js")

for(i in 1:length(files)){

  f <- fromJSON(files[i])
  j <- c(j,f)
}

prev_l <- 1
l <- length(j)
l <- l-prev_l

runNumber <- c(runNumber,rep(265342,1))

files <- list.files(path="C:/Users/gerhard/Documents/msc-thesis-data/processed/000265343", pattern="*.js")

for(i in 1:length(files)){

  f <- fromJSON(files[i])
  j <- c(j,f)
}

prev_l <- 1
l <- length(j)
l <- l-prev_l

runNumber <- c(runNumber,rep(265343,1))

files <- list.files(path="C:/Users/gerhard/Documents/msc-thesis-data/processed/000265344", pattern="*.js")

for(i in 1:length(files)){

  f <- fromJSON(files[i])
  j <- c(j,f)
}

prev_l <- 1
l <- length(j)
l <- l-prev_l

runNumber <- c(runNumber,rep(265344,1))

files <- list.files(path="C:/Users/gerhard/Documents/msc-thesis-data/processed/000265377", pattern="*.js")

for(i in 1:length(files)){

  f <- fromJSON(files[i])

```

```

    j <- c(j,f)
  }

  prev_l <- l
  l <- length(j)
  l <- l-prev_l

  runNumber <- c(runNumber,rep(265377,1))

  files <- list.files(path="C:/Users/gerhard/Documents/msc-thesis-data/processed/000265378", pattern="*.js")

  for(i in 1:length(files)){

    f <- fromJSON(files[i])
    j <- c(j,f)
  }

  prev_l <- l
  l <- length(j)
  l <- l-prev_l

  runNumber <- c(runNumber,rep(265378,1))

  files <- list.files(path="C:/Users/gerhard/Documents/msc-thesis-data/processed/000265381", pattern="*.js")

  for(i in 1:length(files)){

    f <- fromJSON(files[i])
    j <- c(j,f)
  }

  prev_l <- l
  l <- length(j)
  l <- l-prev_l

  runNumber <- c(runNumber,rep(265381,1))

  files <- list.files(path="C:/Users/gerhard/Documents/msc-thesis-data/processed/000265383", pattern="*.js")

  for(i in 1:length(files)){

    f <- fromJSON(files[i])
    j <- c(j,f)
  }

  prev_l <- l
  l <- length(j)
  l <- l-prev_l

  runNumber <- c(runNumber,rep(265383,1))

  files <- list.files(path="C:/Users/gerhard/Documents/msc-thesis-data/processed/000265385", pattern="*.js")

```

```

for(i in 1:length(files)){

  f <- fromJSON(files[i])
  j <- c(j,f)
}

prev_l <- 1
l <- length(j)
l <- l-prev_l

runNumber <- c(runNumber,rep(265385,1))

files <- list.files(path="C:/Users/gerhard/Documents/msc-thesis-data/processed/000265388", pattern="*.js")

for(i in 1:length(files)){

  f <- fromJSON(files[i])
  j <- c(j,f)
}

prev_l <- 1
l <- length(j)
l <- l-prev_l

runNumber <- c(runNumber,rep(265388,1))

files <- list.files(path="C:/Users/gerhard/Documents/msc-thesis-data/processed/000265419", pattern="*.js")

for(i in 1:length(files)){

  f <- fromJSON(files[i])
  j <- c(j,f)
}

prev_l <- 1
l <- length(j)
l <- l-prev_l

runNumber <- c(runNumber,rep(265419,1))

files <- list.files(path="C:/Users/gerhard/Documents/msc-thesis-data/processed/000265420", pattern="*.js")

for(i in 1:length(files)){

  f <- fromJSON(files[i])
  j <- c(j,f)
}

prev_l <- 1
l <- length(j)
l <- l-prev_l

runNumber <- c(runNumber,rep(265420,1))

```

```

files <- list.files(path="C:/Users/gerhard/Documents/msc-thesis-data/processed/000265425", pattern="*.js")

for(i in 1:length(files)){

  f <- fromJSON(files[i])
  j <- c(j,f)
}

prev_l <- 1
l <- length(j)
l <- l-prev_l

runNumber <- c(runNumber,rep(265425,1))

files <- list.files(path="C:/Users/gerhard/Documents/msc-thesis-data/processed/000265426", pattern="*.js")

for(i in 1:length(files)){

  f <- fromJSON(files[i])
  j <- c(j,f)
}

prev_l <- 1
l <- length(j)
l <- l-prev_l

runNumber <- c(runNumber,rep(265426,1))

files <- list.files(path="C:/Users/gerhard/Documents/msc-thesis-data/processed/000265499", pattern="*.js")

for(i in 1:length(files)){

  f <- fromJSON(files[i])
  j <- c(j,f)
}

prev_l <- 1
l <- length(j)
l <- l-prev_l

runNumber <- c(runNumber,rep(265499,1))

rm(runNumber)

Eta <- sapply(j, `[`, "Eta")

P <- sapply(j, `[`, "P")

PT <- sapply(j, `[`, "PT")

Phi <- sapply(j, `[`, "Phi")

Theta <- sapply(j, `[`, "Theta")

```

```

col0 <- sapply(j, `[`, "col0")
col1 <- sapply(j, `[`, "col1")
col2 <- sapply(j, `[`, "col2")
col3 <- sapply(j, `[`, "col3")
col4 <- sapply(j, `[`, "col4")
col5 <- sapply(j, `[`, "col5")
dEdX <- sapply(j, `[`, "dEdX")
det0 <- sapply(j, `[`, "det0")
det1 <- sapply(j, `[`, "det1")
det2 <- sapply(j, `[`, "det2")
det3 <- sapply(j, `[`, "det3")
det4 <- sapply(j, `[`, "det4")
det5 <- sapply(j, `[`, "det5")
nSigmaElectron <- sapply(j, `[`, "nSigmaElectron")
nSigmaPion <- sapply(j, `[`, "nSigmaPion")
pdgCode <- sapply(j, `[`, "pdgCode")
row0 <- sapply(j, `[`, "row0")
row1 <- sapply(j, `[`, "row1")
row2 <- sapply(j, `[`, "row2")
row3 <- sapply(j, `[`, "row3")
row4 <- sapply(j, `[`, "row4")
row5 <- sapply(j, `[`, "row5")
layer0 <- sapply(j, `[`, "layer 0")
layer1 <- sapply(j, `[`, "layer 1")
layer2 <- sapply(j, `[`, "layer 2")
layer3 <- sapply(j, `[`, "layer 3")
layer4 <- sapply(j, `[`, "layer 4")

```

```

layer5 <- sapply(j, `[`, "layer 5")

rm(f,j)

save.image(file="c:/Users/gerhard/documents/msc-thesis-data/all_data.rdata")

load("c:/Users/gerhard/documents/msc-thesis-data/all_data.rdata")

meta <- data.frame(cbind(col0,col1,col2,col3,col4,col5,det0,det1,det2,det3,det4,det5,
                        row0,row1,row2,row3,row4,row5,dEdX,Eta,nSigmaElectron,
                        nSigmaPion,P,pdgCode,PT,Theta,Phi))

save(meta,file="c:/Users/gerhard/documents/msc-thesis-data/meta.rdata")

rm(col0,col1,col2,col3,col4,col5,det0,det1,det2,det3,det4,det5,
   row0,row1,row2,row3,row4,row5,dEdX,Eta,nSigmaElectron,
   nSigmaPion,P,pdgCode,PT,Theta,Phi,f,j)

rm(files,i,pdg.elec,pdg.pion)

n0 <- unique(c(which(sapply(layer0,typeof)== "list"),which(sapply(layer0, is.null)))))
n1 <- unique(c(which(sapply(layer1,typeof)== "list"),which(sapply(layer1, is.null)))))
n2 <- unique(c(which(sapply(layer2,typeof)== "list"),which(sapply(layer2, is.null)))))
n3 <- unique(c(which(sapply(layer3,typeof)== "list"),which(sapply(layer3, is.null)))))
n4 <- unique(c(which(sapply(layer4,typeof)== "list"),which(sapply(layer4, is.null)))))
n5 <- unique(c(which(sapply(layer5,typeof)== "list"),which(sapply(layer5, is.null)))))

save.image(file="c:/Users/gerhard/documents/msc-thesis-data/all_data.rdata")

load("c:/Users/gerhard/documents/msc-thesis-data/all_data.rdata")

meta1 <- meta
meta2 <- meta
meta3 <- meta
meta4 <- meta
meta5 <- meta

rm(meta)

meta0 <- meta1

meta0 <- meta0[-n0,]
layer0 <- layer0[-n0]

meta1 <- meta1[-n1,]
layer1 <- layer1[-n1]

meta2 <- meta2[-n2,]
layer2 <- layer2[-n2]

meta3 <- meta3[-n3,]
layer3 <- layer3[-n3]

meta4 <- meta4[-n4,]
layer4 <- layer4[-n4]

```



```

meta5 <- meta5[-n5,]
layer5 <- layer5[-n5]

rm(n0,n1,n2,n3,n4,n5)

save.image(file="c:/Users/gerhard/documents/msc-thesis-data/all_data.rdata")

load("c:/Users/gerhard/documents/msc-thesis-data/all_data.rdata")

meta0$layer <- 0
meta1$layer <- 1
meta2$layer <- 2
meta3$layer <- 3
meta4$layer <- 4
meta5$layer <- 5

meta0$col <- meta0$col0
meta0$det <- meta0$det0
meta0$row <- meta0$row0

meta1$col <- meta1$col1
meta1$det <- meta1$det1
meta1$row <- meta1$row1

meta2$col <- meta2$col2
meta2$det <- meta2$det2
meta2$row <- meta2$row2

meta3$col <- meta3$col3
meta3$det <- meta3$det3
meta3$row <- meta3$row3

meta4$col <- meta4$col4
meta4$det <- meta4$det4
meta4$row <- meta4$row4

meta5$col <- meta5$col5
meta5$det <- meta5$det5
meta5$row <- meta5$row5

meta0 <- meta0[,-c(1:18)]
meta1 <- meta1[,-c(1:18)]
meta2 <- meta2[,-c(1:18)]
meta3 <- meta3[,-c(1:18)]
meta4 <- meta4[,-c(1:18)]
meta5 <- meta5[,-c(1:18)]

save.image(file="c:/Users/gerhard/documents/msc-thesis-data/all_data.rdata")

load("c:/Users/gerhard/documents/msc-thesis-data/all_data.rdata")

# require(dummies)

meta0 <- as.data.frame(lapply(meta0,unlist))
meta1 <- as.data.frame(lapply(meta1,unlist))

```

```

meta2 <- as.data.frame(lapply(meta2,unlist))
meta3 <- as.data.frame(lapply(meta3,unlist))
meta4 <- as.data.frame(lapply(meta4,unlist))
meta5 <- as.data.frame(lapply(meta5,unlist))

meta <- data.frame(rbind(meta0,meta1,meta2,meta3,meta4,meta5))

# meta <- dummy.data.frame(data=meta,names=list("layer","col","det","row"))
#
# names(meta0)==names(meta1)

rm(meta0,meta1,meta2,meta3,meta4,meta5)

cs0 <- lapply(layer0,colSums)
cs1 <- lapply(layer1,colSums)
cs2 <- lapply(layer2,colSums)
cs3 <- lapply(layer3,colSums)
cs4 <- lapply(layer4,colSums)
cs5 <- lapply(layer5,colSums)

rs0 <- lapply(layer0,rowSums)
rs1 <- lapply(layer1,rowSums)
rs2 <- lapply(layer2,rowSums)
rs3 <- lapply(layer3,rowSums)
rs4 <- lapply(layer4,rowSums)
rs5 <- lapply(layer5,rowSums)

cs0 <- do.call(rbind,cs0)
cs1 <- do.call(rbind,cs1)
cs2 <- do.call(rbind,cs2)
cs3 <- do.call(rbind,cs3)
cs4 <- do.call(rbind,cs4)
cs5 <- do.call(rbind,cs5)

rs0 <- do.call(rbind,rs0)
rs1 <- do.call(rbind,rs1)
rs2 <- do.call(rbind,rs2)
rs3 <- do.call(rbind,rs3)
rs4 <- do.call(rbind,rs4)
rs5 <- do.call(rbind,rs5)

rm(layer0,layer1,layer2,layer3,layer4,layer5)

signal <- data.frame(
  cbind(
    rbind(cs0,cs1,cs2,cs3,cs4,cs5),
    rbind(rs0,rs1,rs2,rs3,rs4,rs5)
  )
)

rm(cs0,cs1,cs2,cs3,cs4,cs5,
  rs0,rs1,rs2,rs3,rs4,rs5)

e <- which(abs(meta$pdgCode)==11)

```

```

p <- which(abs(meta$pdgCode)!=11)

p <- sample(p,length(e),replace = F)

meta <- meta[c(e,p),]
singal <- singal[c(e,p),]

rm(e,p,l,prev_l,singal)

names(meta)

meta$Phi <- scale(meta$Phi)

meta$Theta <- scale(meta$Theta)

meta$PT <- scale(meta$PT)

meta$P <- scale(meta$P)

meta$nSigmaPion <- scale(meta$nSigmaPion)

meta$nSigmaElectron <- scale(meta$nSigmaElectron)

meta$pdgCode <- ifelse(abs(meta$pdgCode)==11,1,0)

meta$dEdX <- scale(meta$dEdX)

meta$Eta <- scale(meta$Eta)

ys <- data.frame(cbind(meta$pdgCode,meta$nSigmaElectron,meta$nSigmaPion))

meta <- meta[,-c(3,4,6)]

names(meta)

require(dummies)

meta <- dummy.data.frame(meta,names=names(meta)[7:10])

names(meta)

singal <- cbind(singal,meta$dEdX,meta$Eta,meta$P,meta$PT,meta$Theta,meta$Phi)

meta <- meta[,-c(1:6)]

load(file="c:/Users/gerhard/documents/msc-thesis-data/all_data.rdata")

singal[,1:41] <- scale(singal[,1:41])

meta <- as.matrix(meta)

singal <- as.matrix(singal)

ys <- as.matrix(ys)

```

```

test_ind <- sample(1:nrow(meta),size=round(0.15*nrow(meta)),replace = F)

meta_test <- meta[test_ind,]

meta <- meta[-test_ind,]

singal_test <- singal[test_ind,]

singal <- singal[-test_ind,]

ys_test <- ys[test_ind,]

ys <- ys[-test_ind,]

require(keras)

rm(meta,meta_test)

ys <- ys[,1]

ys_test <- ys_test[,1]

main_input <- layer_input(shape = c(47), name = 'main_input')
aux_input <- layer_input(shape = c(659), name = 'aux_input')

embed_out <- aux_input %>%
  layer_dense(256,activation = "relu") %>%
  layer_dense(128,activation = "relu") %>%
  layer_reshape(target_shape = c(128))

main_out <- main_input %>%
  layer_dense(256,"relu") %>%
  layer_dense(128,"relu")

merge1 <- layer_concatenate(list(embed_out,main_out)) %>%
  layer_dense(128,"relu") %>%
  layer_dense(2)

merge2 <- layer_concatenate(list(embed_out,main_out)) %>%
  layer_dense(128,"relu") %>%
  layer_dense(1)

model <- keras_model(
  inputs = c(main_input, aux_input),
  outputs = c(merge1,merge2)
)

summary(model)

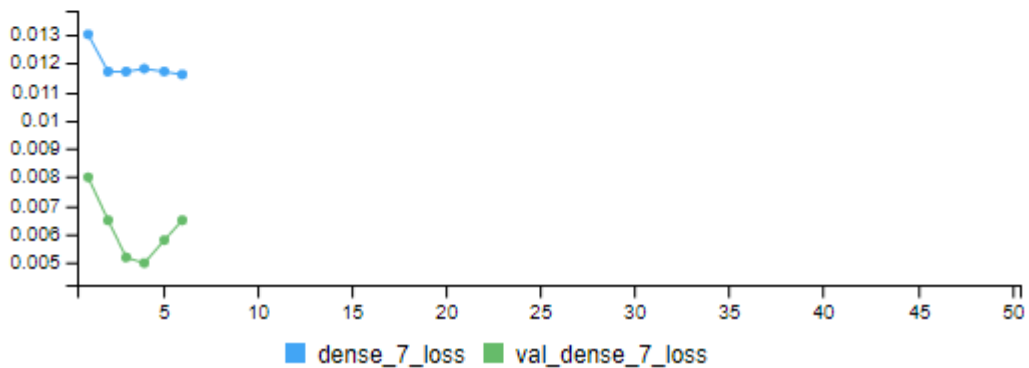
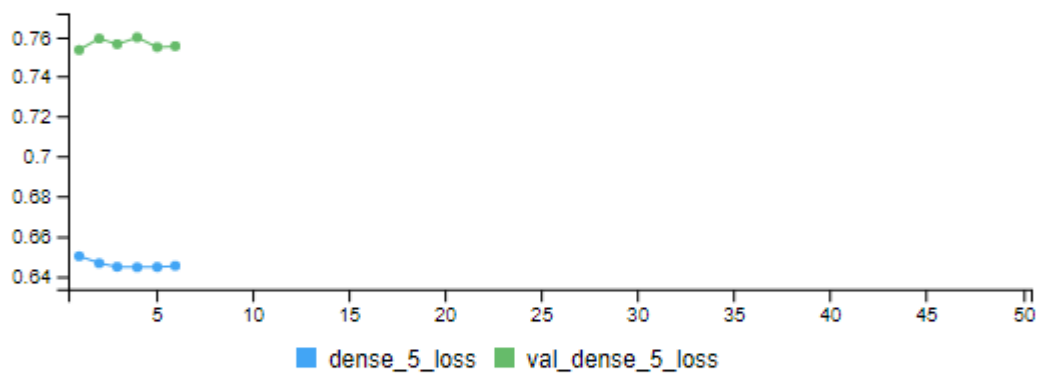
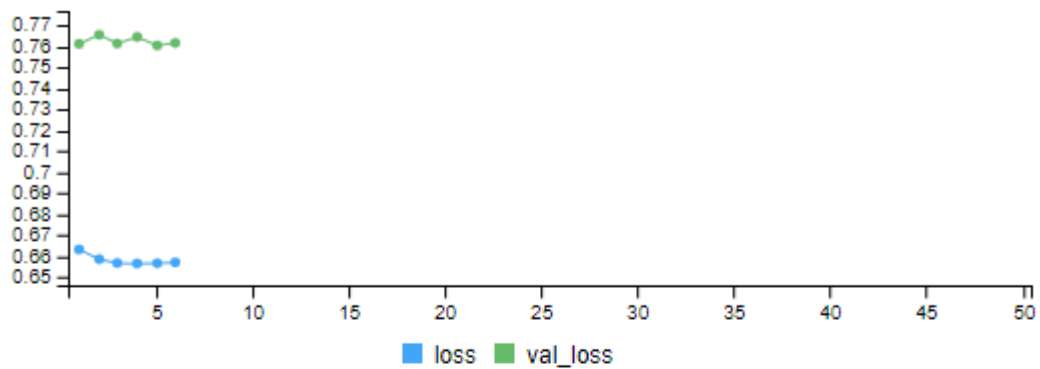
```

```

model %>% compile(
  optimizer = 'rmsprop',
  loss = 'mse'
)

history <- model %>% fit(
  x = list(singal, meta),
  y = list(ys[,2:3],ys[,1]),
  epochs = 50,
  batch_size = 32,
  verbose=2,
  validation_split=0.15,
  metrics=c('mse','acc'),
  shuffle=T
)

```



```

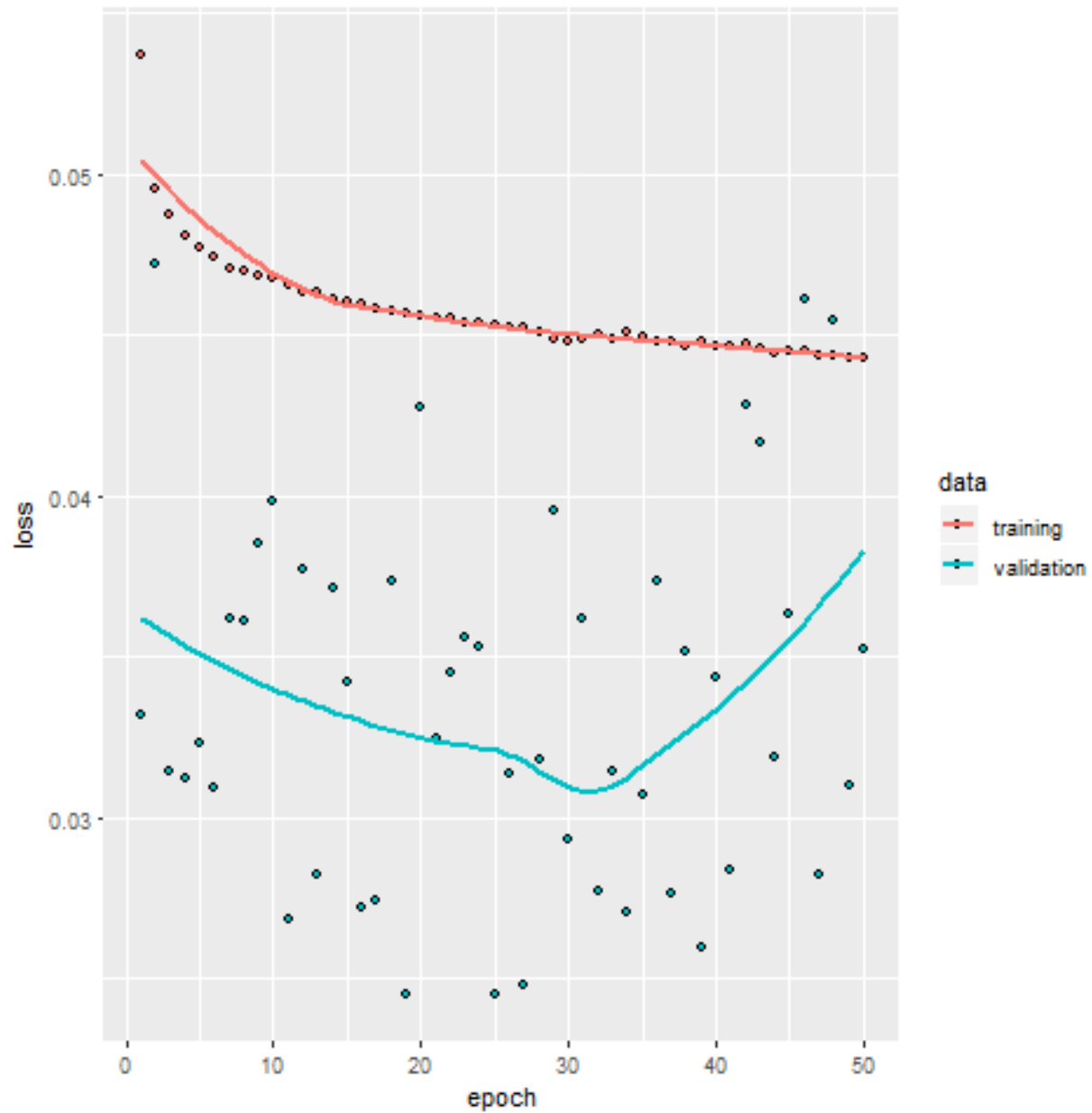
model <- keras_model_sequential() %>%
  layer_dense(256, "tanh") %>%
  layer_dense(128, "tanh") %>%
  layer_dense(1, "sigmoid")

model %>% compile(
  optimizer = 'adam',
  loss = "binary_crossentropy"
)

history <- model %>% fit(
  x = singal,
  y = ys,
  epochs = 50,
  batch_size = 32,
  verbose=2,
  validation_split=0.15,
  metrics=c('acc'),
  shuffle=T
)

png(filename = "C:/Users/gerhard/documents/Msc-thesis/NEW/ML/m2.png")
plot(history)
dev.off()

```



```

pred <- model %>%
  predict(singal_test)

pred_act <- cbind(pred,ys_test)

pred_act <- data.frame(pred_act)

names(pred_act) <- c("pred","act")

pi <- which(pred_act$act==1)
el <- which(pred_act$act!=1)

while(length(pi)%6!=0){
  #print(length(pi))

```

```

pi <- pi[-length(pi)]
}

while(length(el)%%6!=0){
  #print(length(el))
  el <- el[-length(el)]
}

pred_act <- pred_act[c(el,pi),]

six_tracklet_pred <- c()

for(i in seq(1,nrow(pred_act),6)){
  j=i+5

  this.dat <- prod(pred_act$pred[i:j])/sum(prod(pred_act$pred[i:j]),prod(1-pred_act$pred[i:j]))

  six_tracklet_pred <- c(six_tracklet_pred,this.dat)
}

six_tracklet_pred <- data.frame(six_tracklet_pred)

six_tracklet_pred <- na.omit(six_tracklet_pred)

six_tracklet_real <- c()

for(i in seq(1,nrow(pred_act),6)){
  this.dat <- pred_act$act[i]
  six_tracklet_real <- c(six_tracklet_real,this.dat)
}

six_tracklet_real <- data.frame(six_tracklet_real)

which(is.na(six_tracklet_real))

pred_act <- data.frame(cbind(six_tracklet_pred,six_tracklet_real))

elec_pi_eff_func <- function(model_1.preds,model_1.labels){
  # model_1.preds <- read.csv(model_1.preds,header=F, sep="")
  #
  # model_1.labels <- read.csv(model_1.labels,header=F, sep="")

  model_1 <- data.frame(cbind(model_1.preds,model_1.labels))

  model_1.electrons <- which(model_1[,2]==1)

  electrons <- model_1[model_1.electrons,]

  pions <- model_1[-as.numeric(model_1.electrons),]

```



```

electrons <- data.frame(electrons)

names(electrons) <- c("prediction","label")

pions <- data.frame(pions)

names(pions) <- c("prediction","label")

electron_efficiency <- function(electrons.,par){

  electrons <- electrons.

  electrons$electron_pred <- ifelse(electrons$prediction>=par[1],1,0)

  correct <- ifelse(electrons$electron_pred==electrons$label,1,0)

  error_metric <- sum(correct)/nrow(electrons)

  error_metric <- (error_metric-0.9)^2

  return(error_metric)

}

res <- optim(par=c(0),fn=electron_efficiency,lower = 0,upper = 1,electrons.=electrons,method="Brent")

require(ggplot2)

g <- ggplot(pred_act,aes(pred_act[,1],colour=factor(pred_act[,2]))) + geom_histogram(bins = 1000) + facet_wrap(~)
print(g)

hist(pred_act[,1],breaks=1000)
abline(v=res$par,col="red")

electrons$predicted_label <- ifelse(electrons$prediction>=res$par,1,0)

print(paste0("Electron Efficiency: ",sum(electrons$predicted_label)/nrow(electrons)))

pions$predicted_label <- ifelse(pions$prediction>=res$par,1,0)

pions$misclassified_as_electron <- ifelse(pions$predicted_label==1,1,0)

print(paste0("Pion Efficiency: ",(sum(pions$misclassified_as_electron)/nrow(pions))^6))

pred_act$final_pred <- ifelse(pred_act[,1]>=res$par,1,0)

require(caret)

print(caret::confusionMatrix(data=factor(pred_act$final_pred),reference = factor(pred_act[,2])))
print("-----")

}

```

```
elec_pi_eff_func(pred_act[,1],pred_act[,2])
```

```
[1] "Electron Efficiency: 1"
[1] "Pion Efficiency: 0"
Confusion Matrix and Statistics

      Reference
Prediction  0      1
 0 20563      0
 1      0 20496

      Accuracy : 1
      95% CI   : (0.9999, 1)
  No Information Rate : 0.5008
  P-Value [Acc > NIR] : < 2.2e-16

      Kappa : 1

  Mcnemar's Test P-Value : NA

      Sensitivity : 1.0000
      Specificity : 1.0000
   Pos Pred Value : 1.0000
   Neg Pred Value : 1.0000
      Prevalence : 0.5008
   Detection Rate : 0.5008
  Detection Prevalence : 0.5008
   Balanced Accuracy : 1.0000
```

