In Class Assignment 8

Lydia Strebe

April 5, 2019

Training Neural Networks

The following document constructs and analyzes a neural network that has greater than 99% accuracy on the MNIST dataset.

The structure of the neural network, written in Python, was developed using a validation set. Once the validation set consistently had over 99% accuracy, the full training set was used to establish the parameters for the final model.

The last run of my neural network on the validation set returned 99.06% accuracy. Using all the data in the training set returned an accuracy level of 99.26% both within the training set and on the test set.

I believe the results on the test set are higher than the validation set because of the variance-bias trade off. The model couldn't perfectly fit the training data because the accuracy would fall with the addition of new data.

My final neural network is composed of a convolution layer (40 5x5 filters), a maxpooling layer (2x2), two more convolution layers (both 40 filters, the first of size 4x4 and the second of size 3x3), and a final maxpooling layer (2x2). It goes through a flattened layer and then 3 more dense layers with dropouts between each.

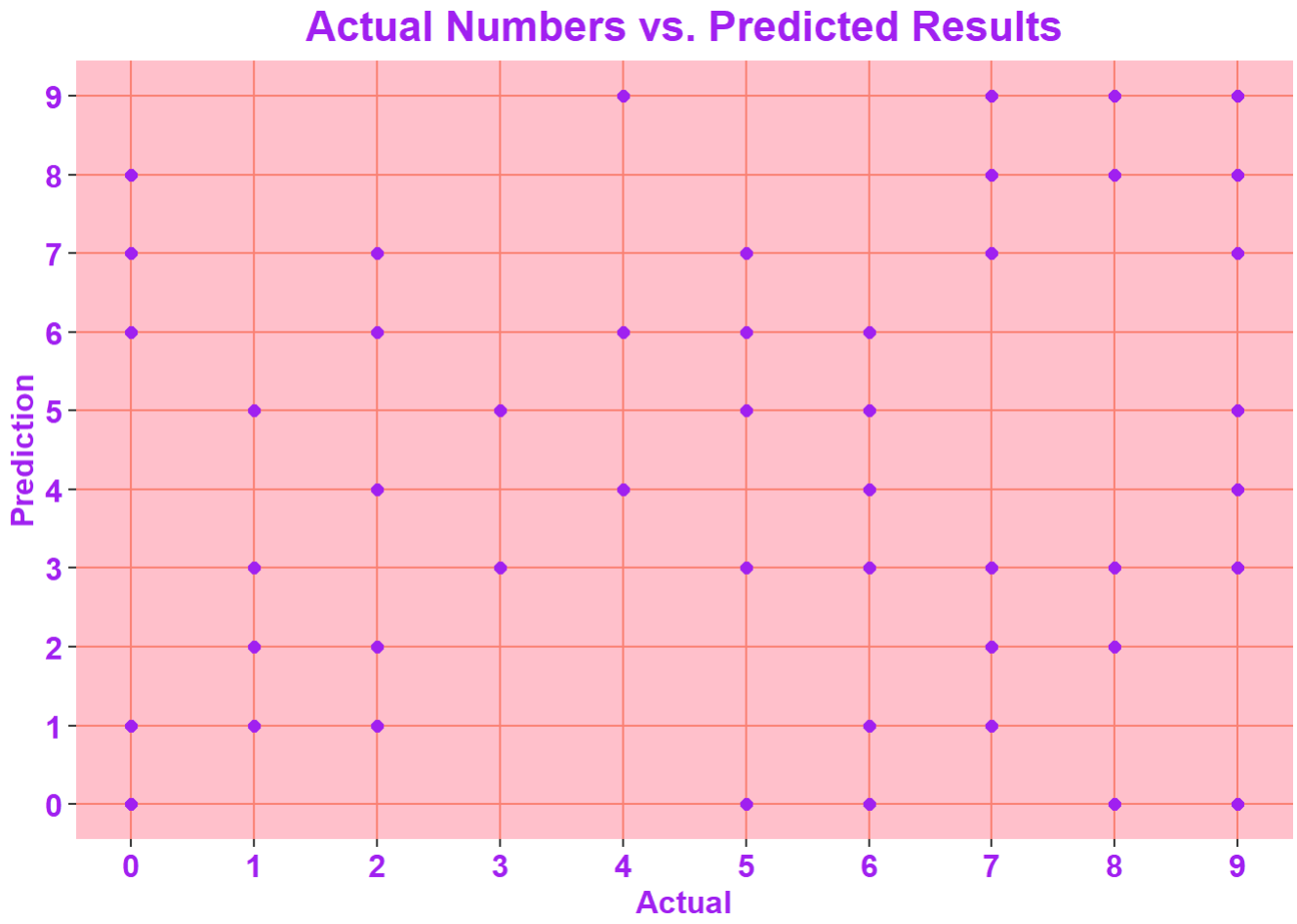
What seemed the most effective at increasing my accuracy was adding convolution layers, increasing the number of filters on my convolution layers, (from 10 to 40), removing regularizers on my dense layers, and adding dropout layers.

I exported the test results to a csv file and plotted the actual values against the predicted values, including some of the numbers that were misclassified.

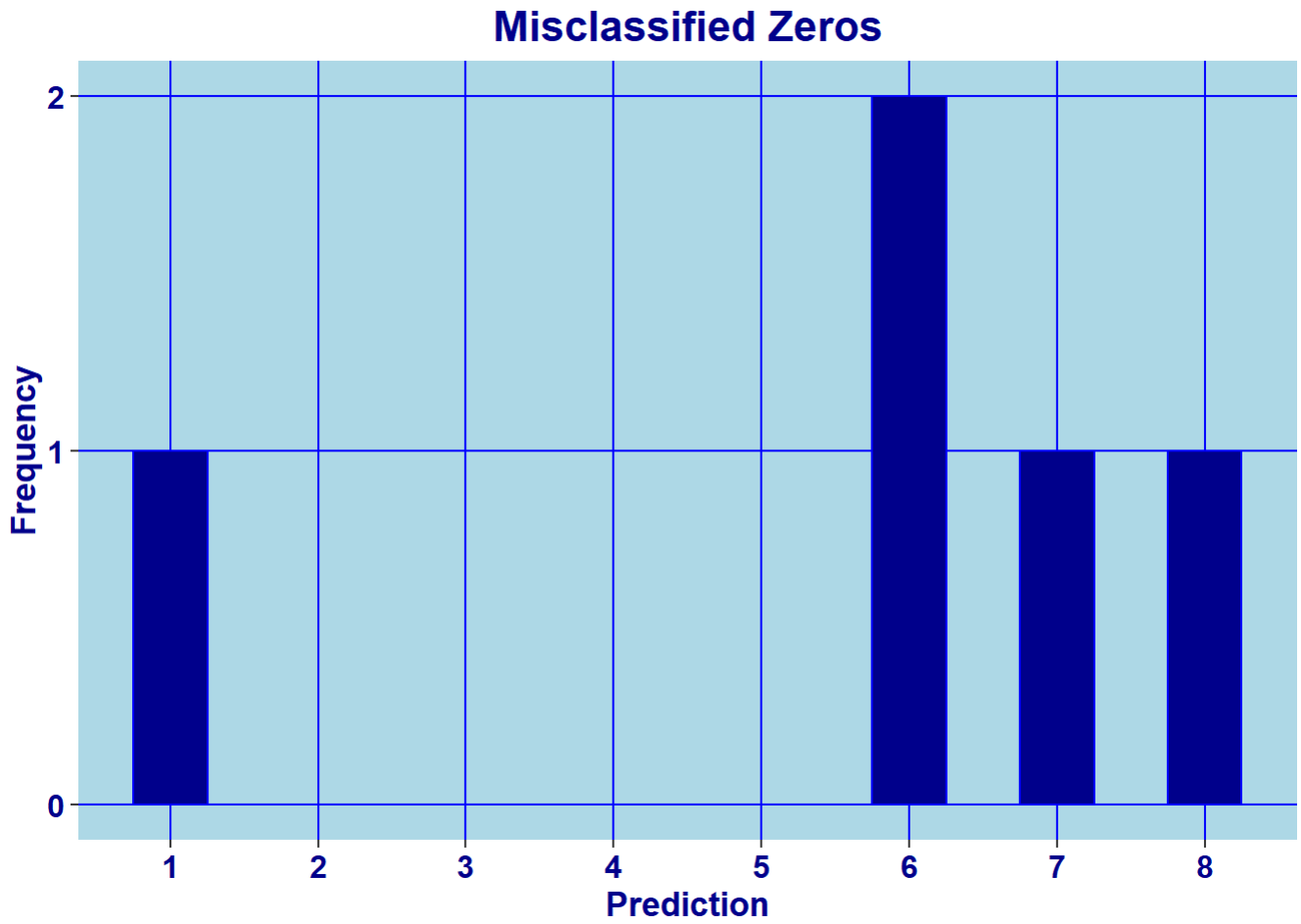
```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.5.2
```

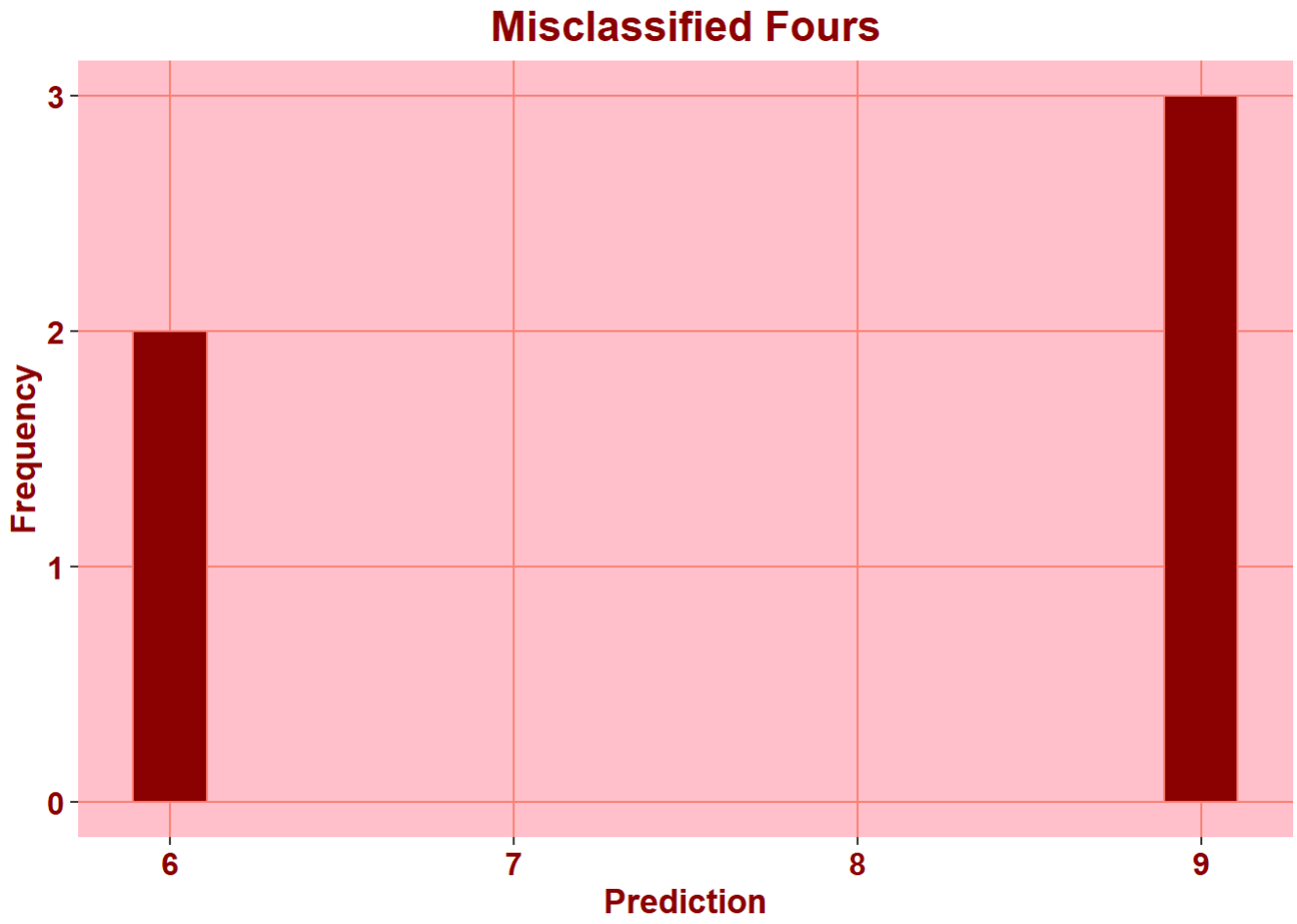
```
class_output=read.csv("C:/Users/lydia/Desktop/Lydia/IE 5561 - Data Driven Decision Making/Assign
ment8/class_output.csv", header = TRUE)

#Plotting all prediction vs. actual
ggplot(data=class_output,aes(y=class_output$prediction,x=class_output$actual))->plot0
plot0+geom_point(size=2,col="purple")+theme(panel.background=element_rect(fill="pink"))->plot1
plot1+labs(title="Actual Numbers vs. Predicted Results",x="Actual",y="Prediction")->plot2
plot2+theme(axis.title=element_text(face="bold",size=12,colour="purple"))->plot3
plot3+theme(plot.title=element_text(hjust=0.5,face="bold",size=16,colour="purple"))->plot4
plot4+scale_x_continuous(breaks=seq(0,10,1))+scale_y_continuous(breaks=seq(0,10,1))->plot5
plot5+theme(panel.grid.major = element_line(colour = "salmon"),panel.grid.minor = element_blank
())->plot6
plot6+theme(axis.text=element_text(face="bold",size=12,colour="purple"))
```



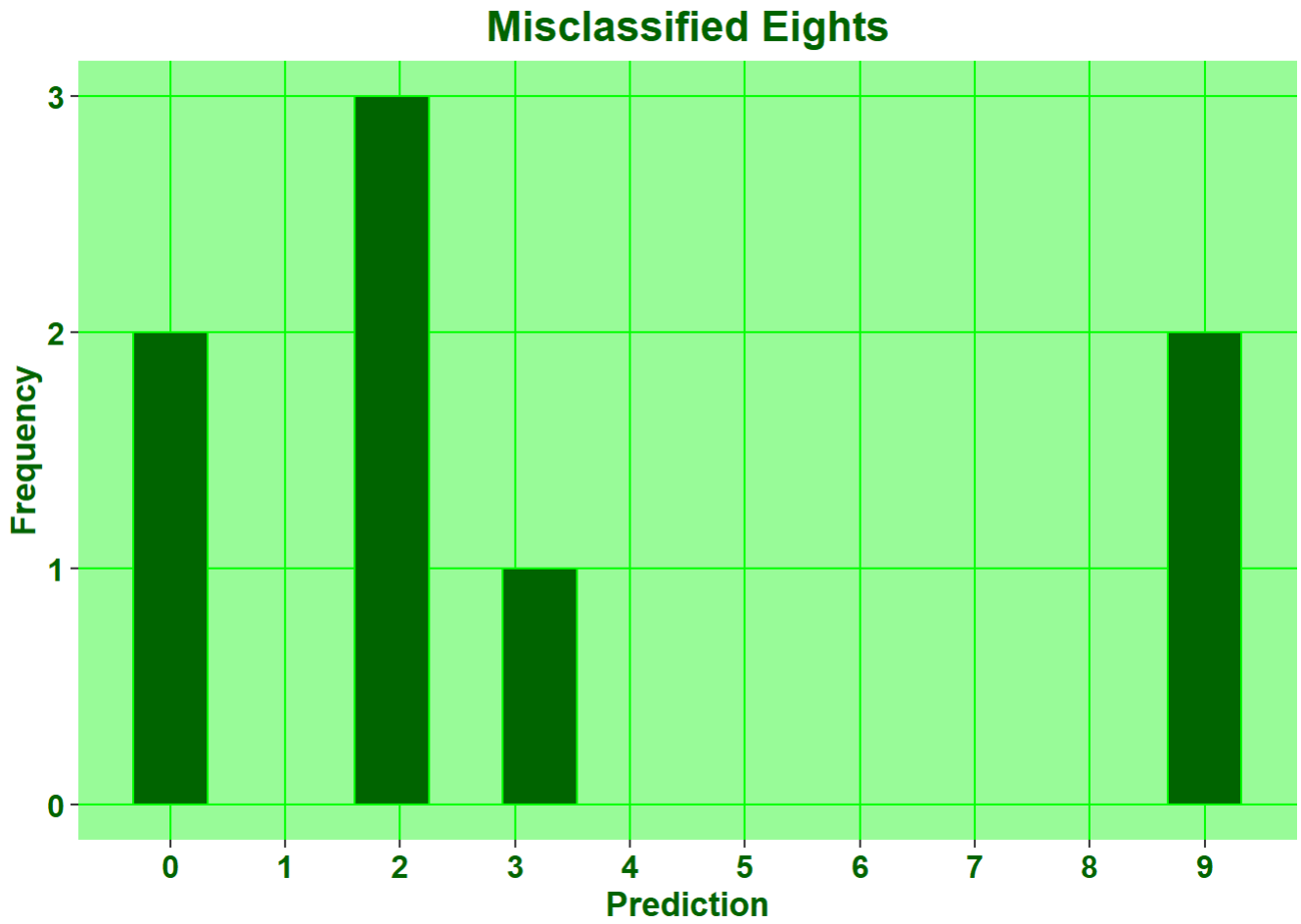

```
#creating data frame of just 0s
zeros=which(class_output$actual == 0)
output_0=data.frame(class_output[zeros,c(1,2)])

#creating data frame of misclassified 0s
not0=which(output_0$prediction != 0)
misclass0=data.frame(output_0[not0,])

#plotting amount/type of misclassified 0s
ggplot(data=misclass0,aes(x=misclass0$prediction))+geom_histogram(bins=15,fill="darkblue",col="b
lue")->hist1
hist1+labs(title="Misclassified Zeros",x="Prediction",y="Frequency")->hist2
hist2+theme(panel.background = element_rect(fill="lightblue"),panel.grid.minor = element_blank
())->hist3
hist3+theme(axis.title=element_text(face="bold",size=13,colour="darkblue"))->hist4
hist4+scale_x_continuous(breaks=seq(0,10,1))+scale_y_continuous(breaks=seq(0,2,1))->hist5
hist5+theme(plot.title = element_text(hjust=0.5,face="bold",size=16,colour="darkblue"))->hist6
hist6+theme(axis.text=element_text(face="bold",size=12,colour="darkblue"))->hist7
hist7+theme(panel.grid.major = element_line(colour = "blue"))
```




```
#creating data frame of just 4s
fours=which(class_output$actual == 4)
output_4=data.frame(class_output[fours,c(1,2)])

#creating data frame of misclassified 4s
not4=which(output_4$prediction != 4)
misclass4=data.frame(output_4[not4,])

#plotting amount/type of misclassified 4s
ggplot(data=misclass4,aes(x=misclass4$prediction))+geom_histogram(bins=15,fill="darkred",col="sa
lmon")->h1
h1+labs(title="Misclassified Fours",x="Prediction",y="Frequency")->h2
h2+theme(panel.background = element_rect(fill="pink"),panel.grid.minor = element_blank())->h3
h3+theme(axis.title=element_text(face="bold",size=13,colour="darkred"))->h4
h4+scale_x_continuous(breaks=seq(0,10,1))+scale_y_continuous(breaks=seq(0,3,1))->h5
h5+theme(plot.title = element_text(hjust=0.5,face="bold",size=16,colour="darkred"))->h6
h6+theme(axis.text=element_text(face="bold",size=12,colour="darkred"))->h7
h7+theme(panel.grid.major = element_line(colour = "salmon"))
```




```
#creating data frame of just 8s
eights=which(class_output$actual == 8)
output_8=data.frame(class_output[eights,c(1,2)])

#creating data frame of misclassified 8s
not8=which(output_8$prediction != 8)
misclass8=data.frame(output_8[not8,])

#plotting amount/type of misclassified 8s
ggplot(data=misclass8,aes(x=misclass8$prediction))+geom_histogram(bins=15,fill="darkgreen",col=
"green")->p1
p1+labs(title="Misclassified Eights",x="Prediction",y="Frequency")->p2
p2+theme(panel.background = element_rect(fill="palegreen"),panel.grid.minor = element_blank())->
p3
p3+theme(axis.title=element_text(face="bold",size=13,colour="darkgreen"))->p4
p4+scale_x_continuous(breaks=seq(0,10,1))+scale_y_continuous(breaks=seq(0,3,1))->p5
p5+theme(plot.title = element_text(hjust=0.5,face="bold",size=16,colour="darkgreen"))->p6
p6+theme(axis.text=element_text(face="bold",size=12,colour="darkgreen"))->p7
p7+theme(panel.grid.major = element_line(colour = "green"))
```



It's not surprising to me that zeros would frequently get mixed up with sixes and eights since these numbers are made up of circles, but I am surprised to see zeros misclassified as sevens and ones.

Fours are a relatively unique-looking number so it makes sense that they don't get confused with a lot of numbers. Nines and fours have a similar structure, so it's not surprising that fours are sometimes misclassified as nines. Additionally, sixes are very similar to nines, except they are upside down, so my neural network must not have fully gotten the orientation right.

I'm not surprised eights get confused with any number with circles or curves, such as zero, three, and nine. I am a little surprised that it wasn't confused with six. I am also surprised by the number of times it was misclassified as a two. However, twos must be tricky for the program to learn since sometimes it is written with a circle and sometimes without.

100% accuracy seems impossible because of the nuances in style and handwriting. E.g. there are two disctinct ways to write a two (with a circle and without) on top of all the differences in handwriting, neatness, orientation, etc. We would not even expect people to get 100% accuracy on so many samples.