Machine Learning: Introduction to Linear Regression, Logistic Regression, and Neural Networks

7.1 Case Study: Spam Classification

Case Study: Spam Classification

Goal of this Section:

- Describe approach for using neural networks for spam classification using the SMS dataset

Text Classification

Text Classification is an application of supervised machine learning
Examples include:

- Spam Classification
  - Binary Classification
  - Training Data: messages and labels (spam or not spam)
  - Goal: predict if new message is spam or not (use to filter email, for example)
- Sentiment Classification of Reviews
  - Multi-class classification
  - Training Data: reviews and labels (1, 2, 3, 4 or 5 star, for example)
  - Goal: predict rating for new reviews

SMS Spam Collection Dataset

- Data located in folder IntroML/Code/Data_Spam
  - readme.txt file provides details about dataset
  - SMSSpamCollection.csv contains the data
- Consist of 5574 text messages: 4827 (not spam) 747 (spam)
- Each line has label (ham or spam) in col A and message in col B

	A	B	C	D	E	F	G	H	I	J	K
1	label	message									
2	ham	Go until jurong point									
3	ham	Ok lar... Joking wif u oni...									
4	spam	Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005. Text FA to 87121 to receive entry ques									
5	ham	U dun say so early hor... U c already then say...									
6	ham	Nah I don't think he goes to usf									
7	spam	FreeMsg Hey there darling it's been 3 week's now and no word back! I'd like some fun you up for it still?									
8	ham	Even my brother is not like to speak with me. They treat me like aids patent.									
9	ham	As per your request 'Melle Melle (Oru Minnaminunginte Nurungu Vettam)' has been set as your callertu									
10	spam	WINNER!! As a valued network customer you have been selected to receivea Â£900 prize reward! To clai									
11	spam	Had your mobile 11 months or more? U R entitled to Update to the latest colour mobiles with camera for									
12	ham	I'm gonna be home soon and i don't want to talk about this stuff anymore tonight									
13	spam	SIX chance	6days	16+ TsandCs apply Reply HL 4 info							
14	spam	URGENT! You have won a 1 week FREE membership in our Â£100									
15	ham	I've been searching for the right words to thank you for this breather. I promise i wont take your help for									
16	ham	I HAVE A DATE ON SUNDAY WITH WILL!!									
17	spam	XXXMobileMovieClub: To use your credit									
18	ham	Oh k...i'm watching here:)									

SMS Spam Collection Dataset - Citation

- Source: (University of California, Irvine, Machine Learning Repository)

http://archive.ics.uci.edu/ml/datasets/SMS+Spam+Collection

- Paper describing work:

Almeida, T.A., Gómez Hidalgo, J.M., Yamakami, A. Contributions to the study of SMS Spam Filtering: New Collection and Results. Proceedings of the 2011 ACM Symposium on Document Engineering (ACM DOCENG'11), Mountain View, CA, USA, 2011. (Under review)

- See website:
  http://www.dt.fee.unicamp.br/~tiago/smsspamcollection/

Converting Messages into a Feature Matrix

- Rudimentary Approach: CountVectorizer (from sklearn package)

1. Build a vocabulary consisting of all words in all messages
   - Not case sensitive ("My" and "my" and "MY" are same word)
2. Feature matrix entry X_{ij} is number of times word i appears in message j
3. Feature Matrix has dimensions (# of words x # of messages)

- Can adjust settings to not include some words (called "stop words"), such as "the", "to", "from", "and", …., which probably do not impact classification.
- See https://scikit-learn.org/stable/index.html for details

CountVectorizer - Example

- 3 Messages: "Call me soon", "CALL to win", "Pick me up soon"
- 7 unique words
- Feature matrix is 7x3

Vocabulary:

$$
\begin{matrix}
call \\
me \\
pick \\
soon \\
to \\
up \\
win
\end{matrix}
$$

Feature Matrix:

$$
\begin{bmatrix}
1 & 1 & 0 \\
1 & 0 & 1 \\
0 & 0 & 1 \\
1 & 0 & 1 \\
0 & 1 & 0 \\
0 & 0 & 1 \\
0 & 1 & 0
\end{bmatrix}
$$

Feature Matrix: Alternative Approaches

- TfIdfVectorizer
  - Tf is acronym for term frequency (how many times a word appears)
  - Idf is acronym for inverse document frequency
  - This function takes into account both number of times a word appears and how many documents in which a word appears
  - See https://scikit-learn.org/stable/index.html and look for TfIdfVectorizer and TfIdfTransformer for details

- Other Approaches
  - CountVectorizer and TfIdfVectorizer don't take into account word orders
  - More sophisticated approaches take into account word order
  - Recurrent Neural Networks

Code Version 4.1: Spam Classification

File/Component	To Do
load_spam	Create function to load spam data
driver_neuralnetwork_spam	Create driver for performing training for spam dataset
text_results	Create function to print false results

New Functions for Spam Classification

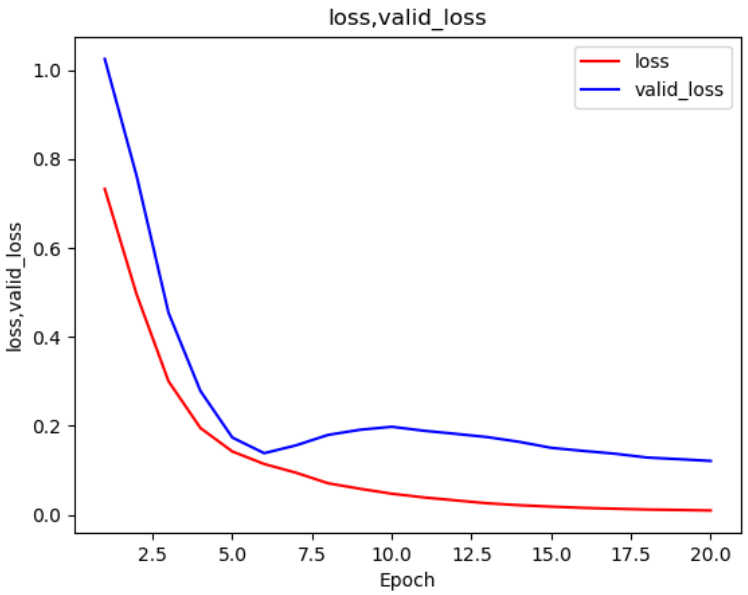
Method	Input	Description
load_spam	ntrain (integer) nvalid (integer)	Loads spam data base and returns train and validation feature matrices (based on CountVectorizer) and label vectors. Also returns original messages in train and validation datasets. Return: Xtrain, Ytrain, Xvalid, Yvalid, Xtrain_raw, Xvalid_raw
data_analysis	X (numpy array) Y (numpy array) nmostcommon (integer) vectorizer (CountVectorizer instance)	Takes in X feature matrix generated by CountVectorizer and label vector Y and prints nmostcommon words in spam and not spam messages Return: nothing
text_results	Y (numpy array) Y_pred (numpy array) X_raw (numpy array)	Given actual Y and predicted Y_pred label vectors and raw messages, this function prints the false positive and false negative messages Return: nothing

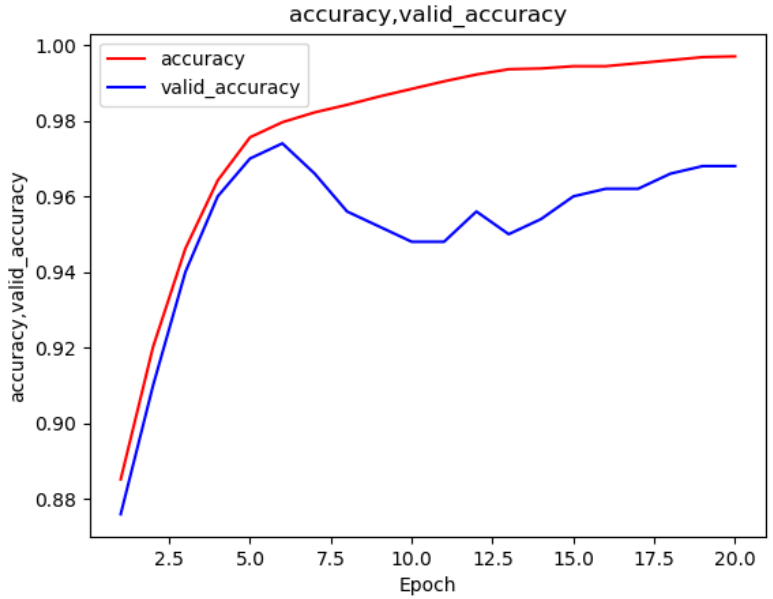
Spam Classification using a Neural Network

- Example: Dataset
  - 5000 messages in training dataset
  - 500 messages in validation dataset
  - 7523 words in vocabulary
  - Feature matrix for training is (7523 x 5000)
  - Feature matrix for validation is (7523 x 500)
- Neural Network
  - 3 layer neural network
    - Layer 1: 200 units (tanh activation)
    - Layer 2: 50 units (tanh activation)
    - Layer 3: 1 unit (sigmoid activation)
    - Binary Cross Entropy Loss function
    - 1,514,901 total entries in $W^{[k]}$ and $b^{[k]}$ for k=1,2,3
- Optimization:
  - Adam
    - α=0.02, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\varepsilon = 10^{-7}$
  - 20 epochs (batch_size = 5000 - batch gradient descent)

Spam Classification – Summary of Results

- After 20 epochs:
  - Training Accuracy: 0.997
  - Validation Accuracy: 0.968
- Loss and Accuracy plots indicate an overfitting





Spam Classification – Summary of Results

- Confusion matrix for validation dataset model predicts
  - 12 False Positive Messages
  - 4 False Negative Messages

```
                Confusion Matrix
                Actual
                0       1
Predicted  0    422     4
           1    12      62
F1Score: 0.8857142850816326 - Precision: 0.8378378372717312 - Recall: 0.9393939386822774
False Positive messages - Actual = not spam - Predicted = spam
My mobile number.pls sms ur mail id.convey regards to achan
Thanks for being there for me just to talk to on saturday. You are very dear to me. I cherish having you as a brother an
d role model.
Hi Shanil
Hi Chikku
Happy new year to u and ur family...may this new year bring happiness
K k:) sms chat with me.
Hi. Hope ur day * good! Back from walk
I.ll hand her my phone to chat wit u
Mode men or have you left.
Dhoni have luck to win some big title.so we will win:)
Have you seen who's back at Holby?!
Also fuck you and your family for going to rhode island or wherever the fuck and leaving me all alone the week I have a
new bong &gt;:(
False Negative messages - Actual = spam - Predicted = not spam
This is the 2nd attempt to contract U
If you don't
dating:i have had two of these. Only started after i sent a text to talk sport radio last week. Any connection do you th
ink or coincidence?
Latest News! Police station toilet stolen
```

Version 4.1: Spam Classification Walkthrough

7.2 MNIST Digits Classification

MNIST Digits Classification

Goal of this Section:

- Describe approach for using neural networks for image classification using the MNIST Digits dataset

Machine Learning - Image Classification

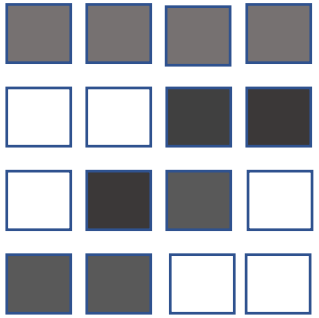
- Image classification (cats and dogs, animals, x-rays, scans, etc) is a principal application of supervised machine learning
- Binary or multi-class
- Training data consists of images plus labels
- Goal is to be able to predict label for new images
- Question: how does one convert images into feature matrix to employ neural network approach?

Representation of Images

- Images typically are composed of rectangular arrays of pixels
- For black and white images, intensity of greyscale for each pixel is represented by a number (white = 0 to 255 = black)
- Feature vector for image is vector of intensities for all pixels
- For colour images, each pixel represented by 3 values – intensities of red, blue, and green components for that pixel – feature vector will have 3 x number of pixels as in black and white case

Converting Image to Feature Matrix

Original Image:
Greyscale 4x4 =16 pixels



Intensity Matrix
4x4 (white=0 to 255=black)

$$\begin{bmatrix} 190 & 190 & 190 & 190 \\ 0 & 0 & 220 & 220 \\ 0 & 220 & 200 & 0 \\ 200 & 200 & 0 & 0 \end{bmatrix}$$

Feature Vector 16x1

$$\begin{bmatrix} 190 \\ 190 \\ 190 \\ 190 \\ 0 \\ 0 \\ 220 \\ 220 \\ 0 \\ 220 \\ 200 \\ 0 \\ 200 \\ 200 \\ 0 \\ 0 \end{bmatrix}$$

Choice of Labels

- For Binary Classification, arbitrarily assign 0, 1 to the classes
  - Example: for classification of cats and dogs (assign 0 for cat and 1 for dog)
  - Example: X-rays assign (0 normal and 1 for broken)
  - Choice is arbitrary (can use 1 for cat and 0 for dog) – doesn't matter
- For Multiclass Classification (c classes) assign 0,1,…,c-1 to classes
  - For digits classification, 10 classes – obviously assign 0 to 0, 1 to 1, …, 9 to 9
  - For pictures of cats, dogs, rabbits, ferrets, ducks (5 classes), assign 0 to cats, 1 to dogs, 2 to rabbits, 3 to ferrets, and 4 to ducks.

MNIST Digits Database

- NIST is acronym for National Institute of Standards and Technology, which is a physical sciences laboratory and a non-regulatory agency of the United States Department of Commerce
- MNIST (Modified National Institute of Standards and Technology) digits database is a large collection of black and white handwritten digit images used for training and testing of machine learning algorithms
- Digit images are uniform (28x28 resolution = 784 pixels)
- 60,000 individual digit images (0 – 9 with labels) for training
- 10,000 individual digit images (0 – 9 with labels) for testing
- Data Source: http://yann.lecun.com/exdb/mnist/

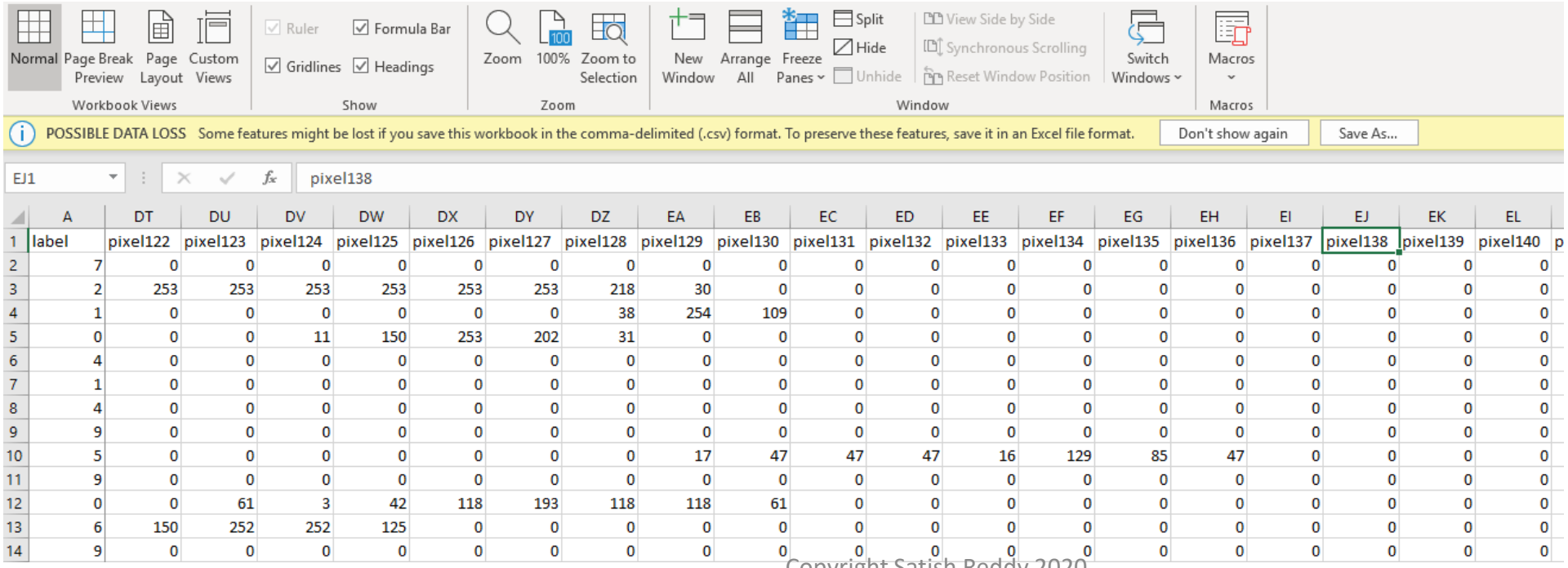
Sample of Digit Images

- Collage of 160 individual digit images
- Citation for above image

By Josef Steppan - Own work, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=64810040

MNIST Digits – Format of Data Files

- Each row represents label and intensities for one image
  - First column is the label (0,1,…,9)
  - Columns 2 – 785 are the intensities
  - Take transpose before inputting before training
  - Standard practice is to divide pixel values by 255 so between 0 and 1



	A	DT	DU	DV	DW	DX	DY	DZ	EA	EB	EC	ED	EE	EF	EG	EH	EI	EJ	EK	EL
1	label	pixel122	pixel123	pixel124	pixel125	pixel126	pixel127	pixel128	pixel129	pixel130	pixel131	pixel132	pixel133	pixel134	pixel135	pixel136	pixel137	pixel138	pixel139	pixel140
2	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	2	253	253	253	253	253	253	218	30	0	0	0	0	0	0	0	0	0	0	0
4	1	0	0	0	0	0	0	38	254	109	0	0	0	0	0	0	0	0	0	0
5	0	0	0	11	150	253	202	31	0	0	0	0	0	0	0	0	0	0	0	0
6	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	5	0	0	0	0	0	0	0	17	47	47	47	16	129	85	47	0	0	0	0
11	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	61	3	42	118	193	118	118	61	0	0	0	0	0	0	0	0	0	0
13	6	150	252	252	125	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Training and Validation Data

- Data located in folder IntroML/Code/Data_MNIST:
  - 60000 training data samples split into 2 files (because of Github limitations)
    - MNIST_train_set1_30K.csv
    - MNIST_train_set2_30K.csv
    - 1 data sample for each line consisting of digit label plus 784=28x28 pixel values
  - 10000 validation data samples in file:
    - MNIST_valid_10K.csv
    - 1 data sample for each line consisting of digit label plus 784=28x28 pixel values

Coding Walkthrough: Version 4.1 To Do

File/Component	To Do
load_mnist	Create function to load MNIST data
driver_neuralnetwork_mnist	Create driver for performing training for MNIST dataset
plot_results_mnist_animation	Create function to provide animation of digits

New Functions for Digits Classification

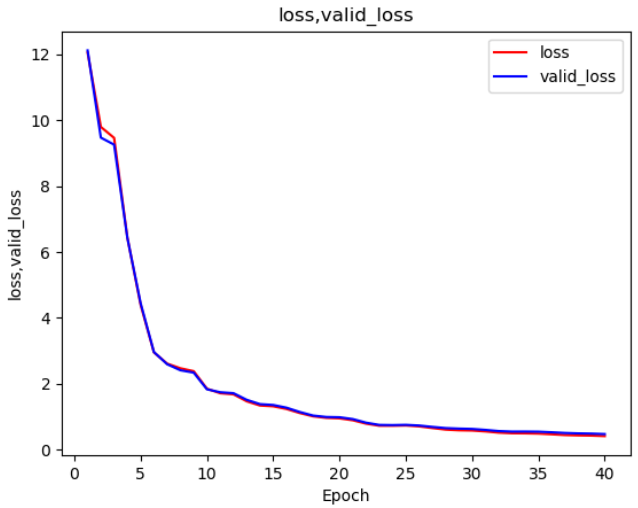
Method	Input	Description
load_mnist	ntrain (integer) nvalid (integer)	Loads MNIST database Return: Xtrain, Ytrain, Xvalid, Yvalid
plot_results_mnist_animation	X (numpy array) Y (numpy array) Y_pred (numpy array) nframe (integer)	Shows animation of digit images (X) and prints actual label (Y) and predicted label (Y_pred) for nframe images Return: nothing

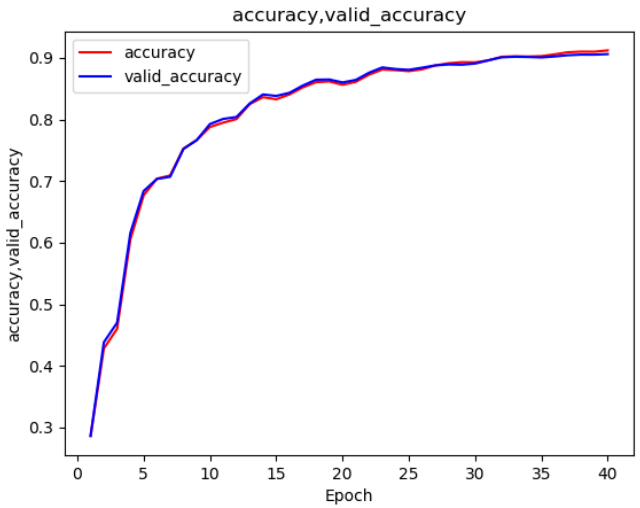
MNIST Digit Classification using a Neural Network

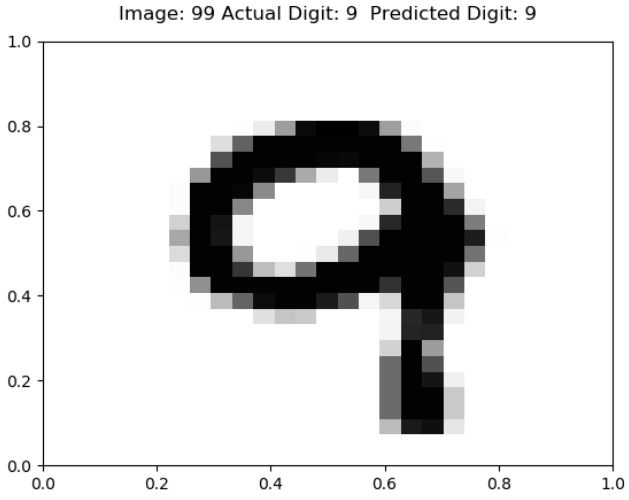
- Example: Dataset
  - 60000 images (28x28 resolution) in training dataset
  - 10000 images in validation dataset
  - Feature matrix for training is (784 x 60000)
  - Feature matrix for validation is (784 x 10000)
- Neural Network
  - 2 layer neural network
    - Layer 1: 128 units (tanh activation)
    - Layer 2: 10 unit (softmax activation)
    - Cross Entropy Loss function
    - 101,770 total entries in $W^{[k]}$ and $b^{[k]}$ for k=1,2
- Optimization:
  - Adam
    - α=0.02, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\varepsilon = 10^{-7}$
    - 40 epochs (batch_size = 60000 batch gradient descent)

Digit Classification – Summary of Results

- After 40 epochs:
  - Training Accuracy: 0.912
  - Validation Accuracy: 0.906
- Loss and Accuracy plots indicate an underfitting
  - Expect training accuracy to be higher – should be close to 100%







Digit Classification – Summary of Results

- Confusion Matrix shows
  - Most difficulty predicting digits 5 and 8
  - Digits 4 and 9 often mistaken for each other

Confusion Matrix

		Actual									
		0	1	2	3	4	5	6	7	8	9
Predicted	0	938	0	10	3	1	19	8	0	19	10
	1	0	1103	4	3	1	1	2	6	4	2
	2	6	5	925	26	9	5	6	29	23	3
	3	5	9	36	919	2	48	2	17	45	15
	4	1	1	12	2	909	18	7	6	13	43
	5	4	2	0	17	2	738	14	1	27	3
	6	17	6	11	1	11	10	915	0	14	3
	7	3	0	17	16	2	18	1	908	16	17
	8	1	9	14	15	2	28	2	4	798	5
	9	5	0	3	8	43	7	1	57	15	908

Version 4.1: Digits Classification Walkthrough