Digital Reasoning

Deep Learning for Text Classification

Joseph Porter, Text Research

September 9, 2019

Part 1: Fun with NLP!

https://talktotransformer.com/

Digital Reasoning

Natural Language Processing (NLP)

“The global Natural Language Processing (NLP) market accounted to hit the market value of **US$ 28.6 Bn in 2026** and is expected to witness **CAGR of 11.71%** across the forecast period through 2018 to 2026.”

- Research and Markets, March 2019

*CAGR = Compound Annual Growth Rate

Digital Reasoning

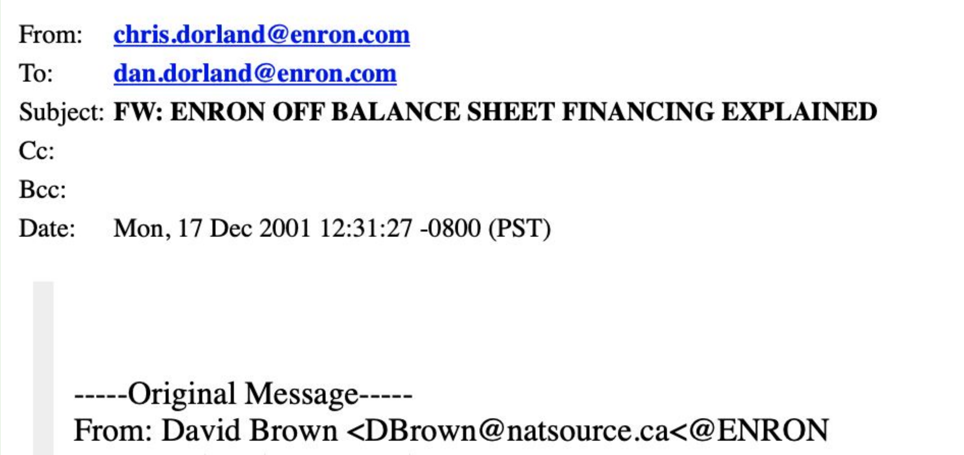
What is NLP?

Unstructured Text

Lots of e-mail and chat
Customer feedback
Transcribed audio

Semistructured Text

Logs
Medical records
Transaction transcripts



From: chris.dorland@enron.com
To: dan.dorland@enron.com
Subject: **FW: ENRON OFF BALANCE SHEET FINANCING EXPLAINED**
Cc:
Bcc:
Date: Mon, 17 Dec 2001 12:31:27 -0800 (PST)

-----Original Message-----
From: David Brown <DBrown@natsource.ca<@ENRON

Digital Reasoning

What can we do with NLP?

CLASSIFICATION

User intent detection for sales or support

Categorizing customer feedback

Customer complaint intensity

Public sentiment towards our company or competitors

INFORMATION EXTRACTION

Parsing semi-structured data:

- Cancer reports
- Equities trades
- ...

SIMILARITY

Semantic text search

Smart queries

Recommendation systems

ADVANCED - AI-Type Tasks

Machine Translation

Text generation

Question answering

Chatbots

Assertions / Reasoning

Digital Reasoning

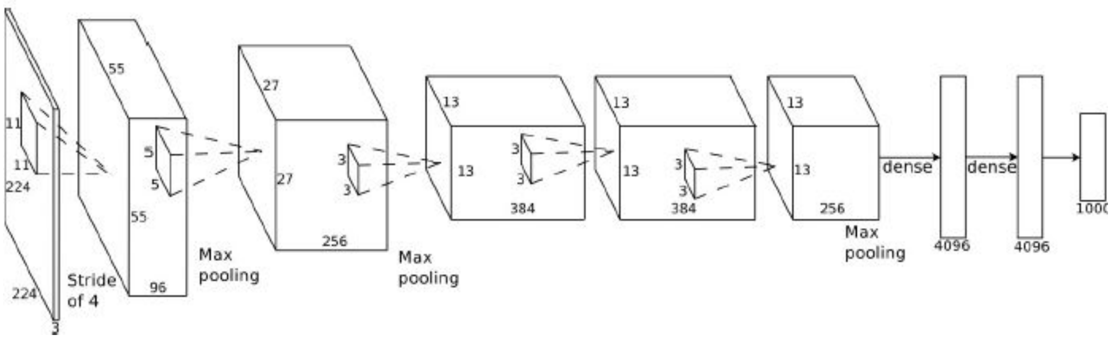
Deep Learning and Text Transfer

Learn information from one task and apply it to another task

Transfer learning methods were pioneered in image classification

Our model

- Max-pooling layers follow first, second, and fifth convolutional layers
- The number of neurons in each layer is given by 253440, 186624, 64896, 64896, 43264, 4096, 4096, 1000



http://www.image-net.org/challenges/LSVRC/2012/supervision.pdf
http://cs231n.github.io/transfer-learning/

Deep Learning and Text Transfer

Learn information from one task and apply it to another task

Transfer learning methods were pioneered in image classification, **and were successfully applied to text!**

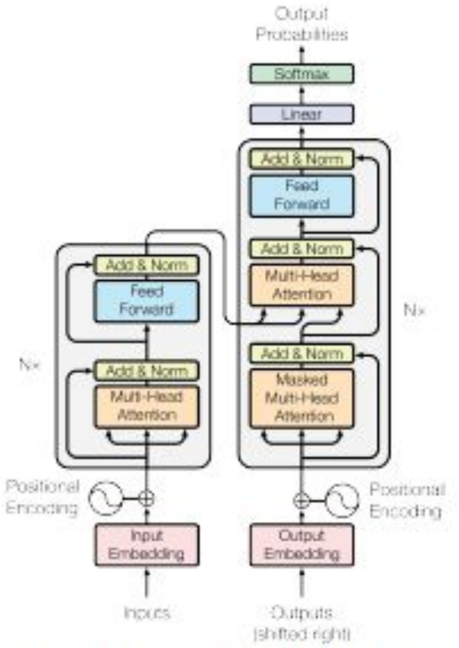
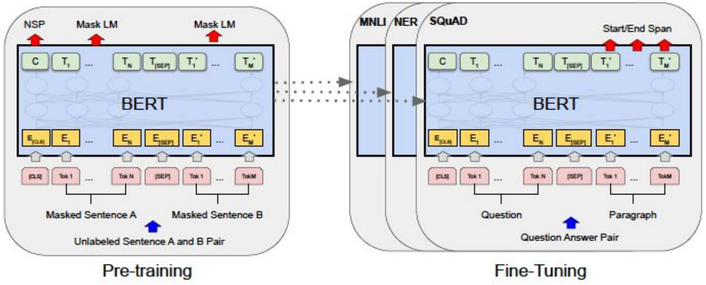


Figure 1: The Transformer - model architecture.

Vaswani et al. Attention is All You Need. https://arxiv.org/abs/1706.03762



Pre-training w/ unsupervised data

Fine-tuning for particular supervised tasks

Devlin et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. https://arxiv.org/abs/1810.04805

Digital Reasoning

Why Deep Learning?

- Reduce the burden on the data scientist
    - Learn the features instead of experimenting / coding - remove one dimension from the problem
    - Transfer learning gets us up and running with less time and less data
- Achieve much higher performance

Potential Costs

- Bigger, slower models
- Need for GPUs
- More data needed (sometimes)
- Know-how required

Part 2: Classifiers

Code for this adventure is available here: https://github.com/uphill-ai/NAS2019

Official Disclaimers:

There are many more steps involved in making sure you have built a good model usable for production (data cleaning, preprocessing, model selection, hyperparameter optimization, careful evaluation, etc…).

Things we'll cover today:

- How to think about a deep learning transfer model.
- How to get set up and make sure you're on the right track.
- Quickly getting a decent prototype for experimentation & business validation.

Caveats:

- I used bert-sklearn, which has some limitations - the API is easy to follow.
- I would rather use pytorch-transformers. I have some versions of the notebook code which I can clean up and make available via github.

Deep Learning Text Classifier Recipe

Ingredients

- Text Data
- Labels
- Pretrained Model
- Parameters
- Computer! (GPU is best)

Steps

1. Clean up data and subdivide
2. Choose initial parameters
3. Iterate to:
    a. Resolve Problems
    b. Improve Quality
    c. Avoid Overfitting
4. Deploy your model somewhere

Deep Learning Text Classifier Recipe

Ingredients

- Text Data
- Labels
- Pretrained Model
- Parameters
- Computer! (GPU is best)

Steps

1. Clean up data and subdivide
2. Choose initial parameters
3. Iterate to:
    a. Resolve Problems
    b. Improve Quality
    c. Avoid Overfitting
4. Deploy your model somewhere

Data!

CrowdFlower (now Figure Eight)
Brands and Product Emotions

Apple / Google Products
Tweets with Sentiment Labels
Positive, Negative, or Neutral

8721 Examples, Quality good

http://data.world/crowdflower/brands-and-product-emotions

Kaggle
Sentiment 140

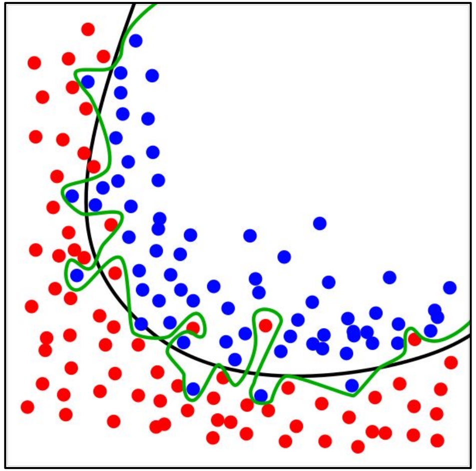
Tweets with Sentiment Labels

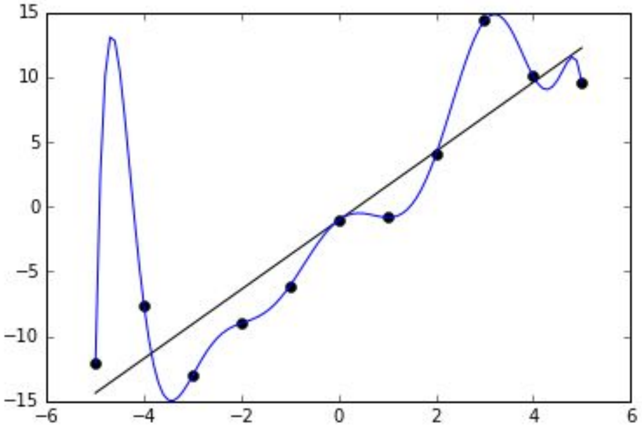
Positive or Negative only

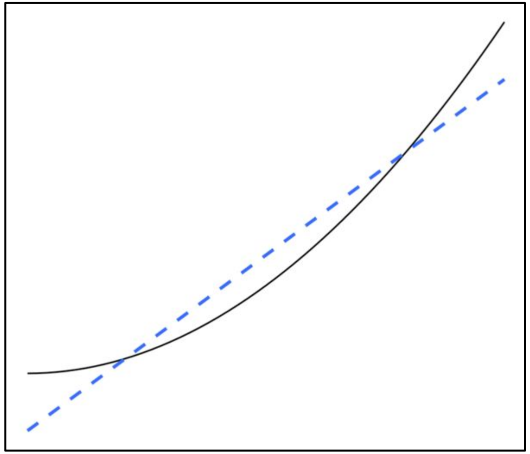
1.6M Examples, Quality fair

https://www.kaggle.com/kazanova/sentiment140

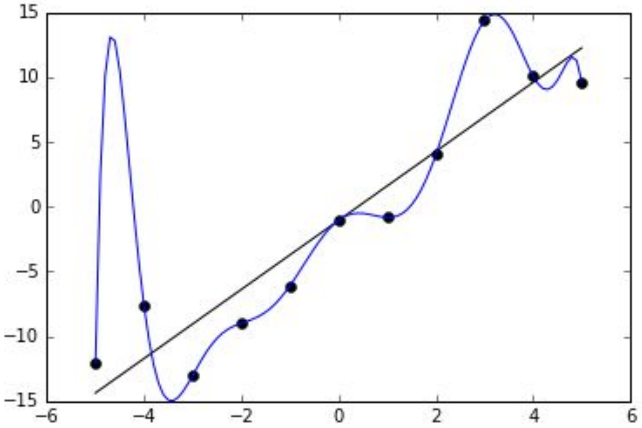
Overfitting (thanks, Wikipedia)





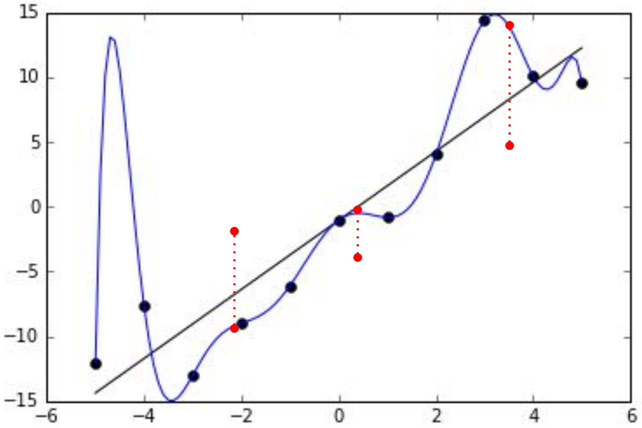


Overfitting (thanks, Wikipedia)



What happens when we predict with an overfit model?
Add some new points and see!

Overfitting (thanks, Wikipedia)



ALL MISSES!!!

What happens when we predict with an overfit model?
Add some new points and see!

The whole point of machine learning is to create a representation that generalizes to data that it hasn't seen before, and makes valid predictions.

Digital Reasoning

Ensuring Generalization

Snooping: every piece of test data can be used either for <u>optimization</u> **OR** <u>evaluation</u>, **BUT NOT BOTH**

Solution: Need a final holdout set for evaluation, and a validation set for optimization. Randomly split our data into three pieces (train, test, validation).

https://stats.stackexchange.com/questions/107617/how-is-cross-validation-different-from-data-snooping

https://stats.stackexchange.com/questions/312404/general-strategy-for-tuning-a-ml-algorithm

Ensuring Generalization

Snooping: every piece of test data can be used either for optimization **OR** evaluation, **BUT NOT BOTH**

Solution: Need a final holdout set for evaluation, and a validation set for optimization. Randomly split our data into three pieces (train, test, validation).

Question: So why are you snooping in your notebooks?

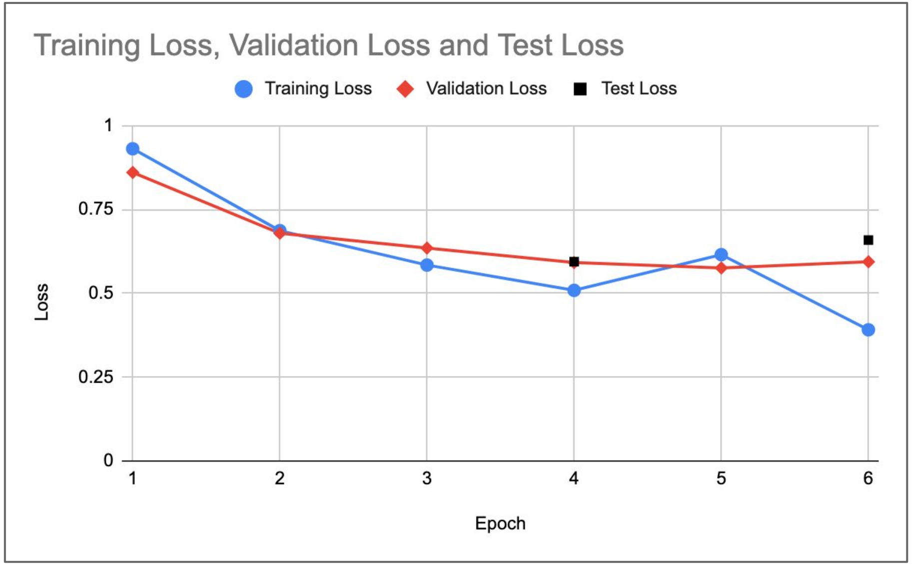
Answer: This is a pilot (and I'm only snooping a little) !!!

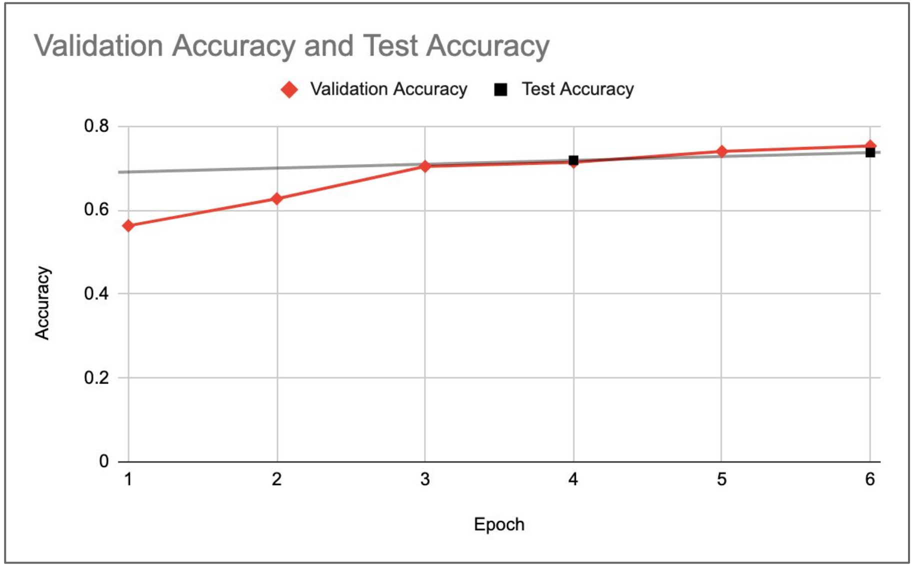
https://stats.stackexchange.com/questions/107617/how-is-cross-validation-different-from-data-snooping

https://stats.stackexchange.com/questions/312404/general-strategy-for-tuning-a-ml-algorithm

Detecting Overfit with Learning Curves

… and Preventing Overfitting with Early Stopping





Digital Reasoning

BERT Hyperparameters (a few of them)

Parameter	Typical Values	Where Used	Notes
Model name	bert-base-uncased	Model Selection	Which pretrained model? BERT regular size, no case
max_seq_length	128, 256, 512	Data Mgmt	Each input is truncated to this length (in words). Max is 512.
learning_rate	1e-5, 2e-5, 4e-5	Training	More to follow! How fast to step the optimization
epochs	3,4,10,20	Training	More to follow! How many times to train on the input data
batch_size	16,32,64	Data Mgmt	How many examples do we pass to the GPU at a time?
vocab_size	30000	Data mgmt	Size of the vocabulary set
hidden_dropout_prob	0.1	Training	Regularization parameter
num_hidden_layers	12	Model Arch	Depth of model (don’t change this! :-))

BERT Hyperparameters (a few of them)

Parameter	Typical Values	Where Used	Notes
Model name	bert-base-uncased	Model Selection	Which pretrained model? BERT regular size, no case
max_seq_length	128, 256, 512	Data Mgmt	Each input is truncated to this length (in words). Max is 512.
learning_rate	1e-5, 2e-5, 4e-5	Training	More to follow! How fast to step the optimization
epochs	3,4,10,20	Training	More to follow! How many times to train on the input data
batch_size	16,32,64	Data Mgmt	How many examples do we pass to the GPU at a time?
vocab_size	30000	Data mgmt	Size of the vocabulary set
hidden_dropout_prob	0.1	Training	Regularization parameter
num_hidden_layers	12	Model Arch	Depth of model (don't change this! :-))

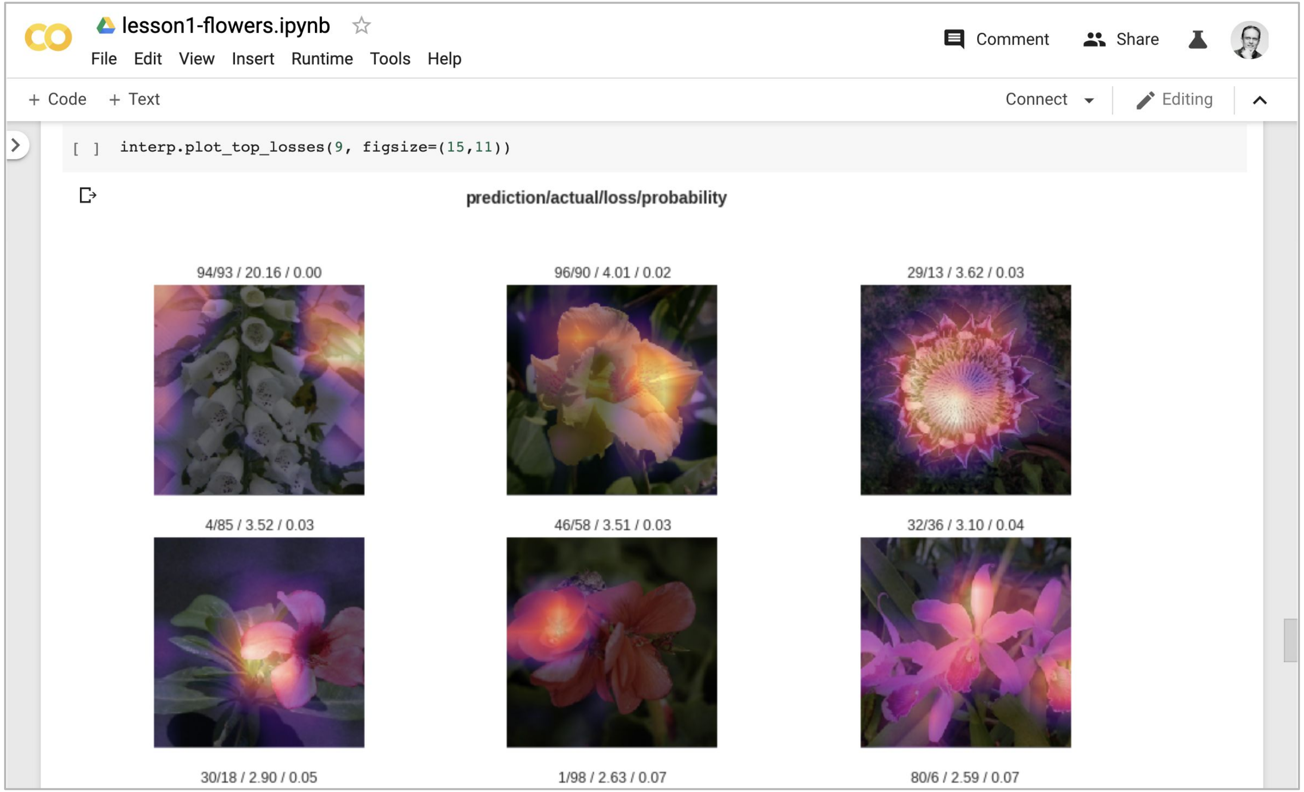
Training time is precious!

Training time is precious!

How do we make sure to use our time wisely?

1. Start with a very small subset of your data
2. Make sure all of the steps run with small data
3. Step up to <u>larger</u> data to experiment
4. Use a GPU!!!
   a. If you have one, great! (if you can get it set up)
   b. Cloud GPUs (beware of leaving them on!)
   c. Google Colaboratory!!!

Google Colaboratory (Google it!)



Digital Reasoning

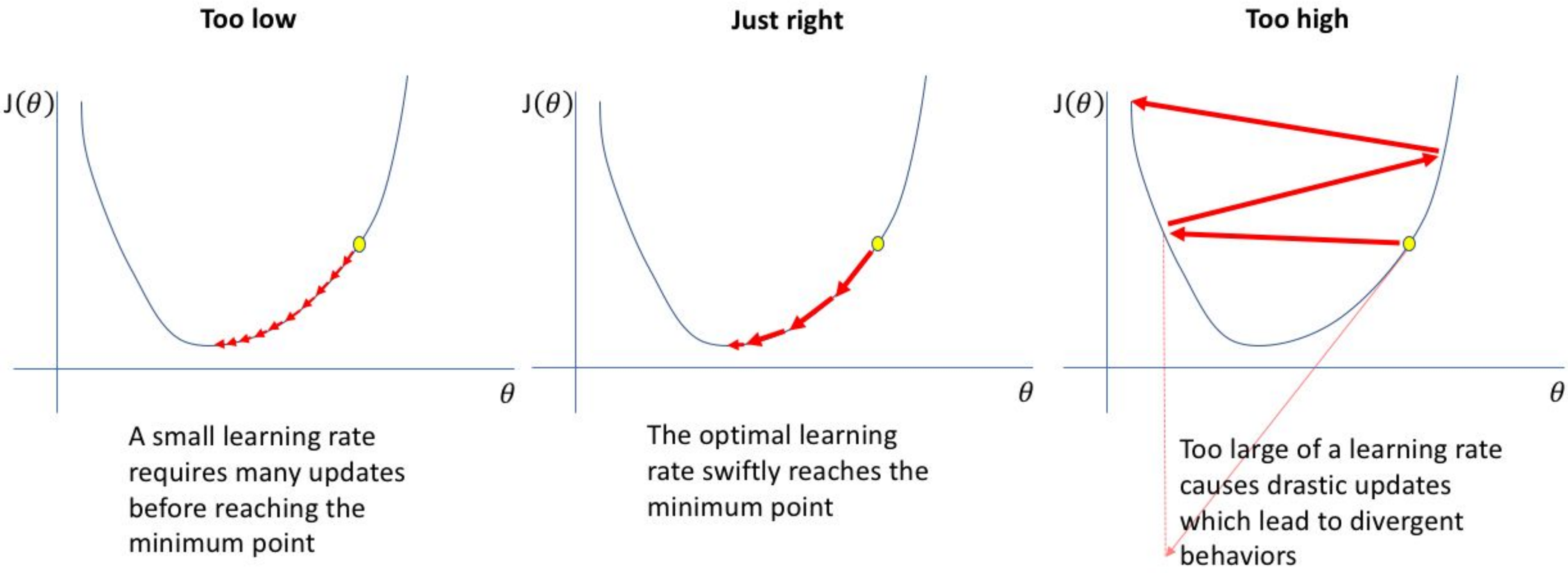
BERT Hyperparameters (a few of them)

Parameter	Typical Values	Where Used	Notes
Model name	bert-base-uncased	Model Selection	Which pretrained model? BERT regular size, no case
max_seq_length	128, 256, 512	Data Mgmt	Each input is truncated to this length (in words). Max is 512.
learning_rate	1e-5, 2e-5, 4e-5	Training	More to follow! How fast to step the optimization
epochs	3,4,10,20	Training	More to follow! How many times to train on the input data
batch_size	16,32,64	Data Mgmt	How many examples do we pass to the GPU at a time?
vocab_size	30000	Data mgmt	Size of the vocabulary set
hidden_dropout_prob	0.1	Training	Regularization parameter
num_hidden_layers	12	Model Arch	Depth of model (don’t change this! :-))

Digital Reasoning

Training time is precious!

Learning Rate



Demo 2: Classifiers

Trial Run Results

How to Measure Results

Accuracy: How many did we get right (in %)?

Problem: What if the data are severely imbalanced?

My awesome binary classifier: 80% accuracy

… BUT …

The data has 88% yes and 12% no.

So if I predict yes all the time, my accuracy goes up to 88% !!!

Maybe accuracy isn’t the best thing here.

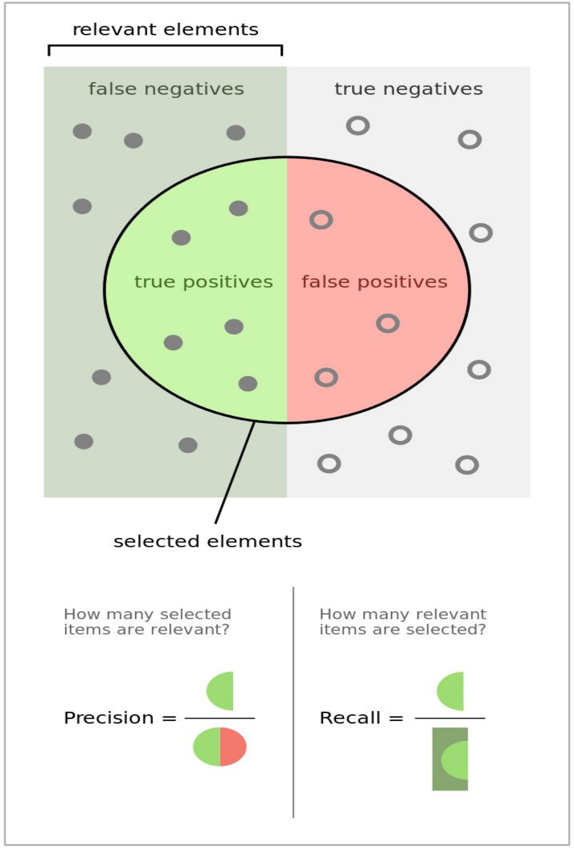
How to Measure Results, part 2

Question. If accuracy is so easy to fool, what do we use?

Answer: Precision and Recall !!!

Recall: Of the samples that were positive, what fraction did the model find?

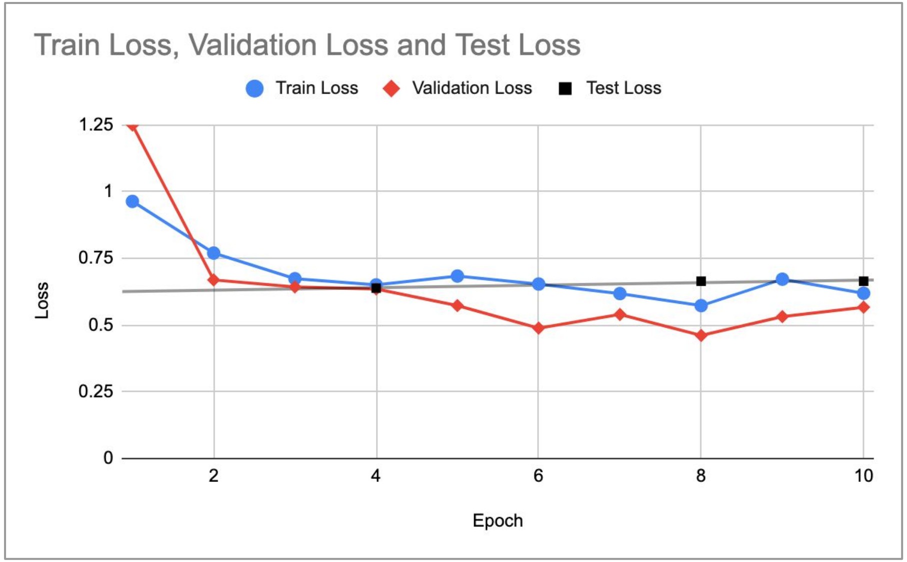
Precision: Of the samples that were found, what fraction were correct?

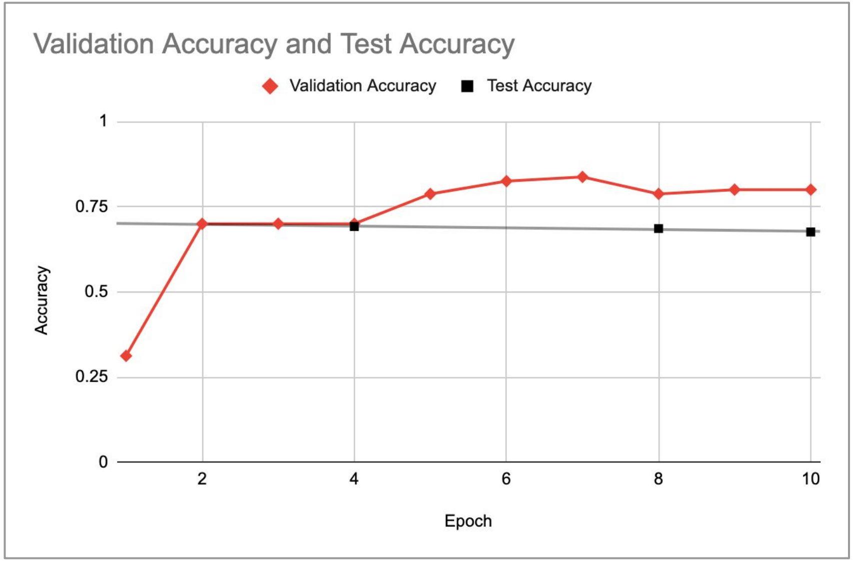


https://en.wikipedia.org/wiki/Precision_and_recall

Digital Reasoning

Model Diagnostics (self-justification for snooping)

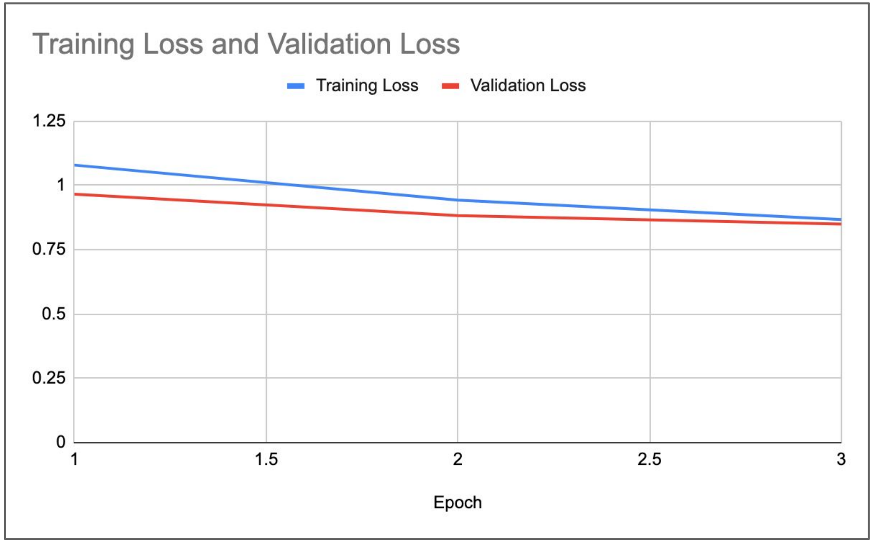




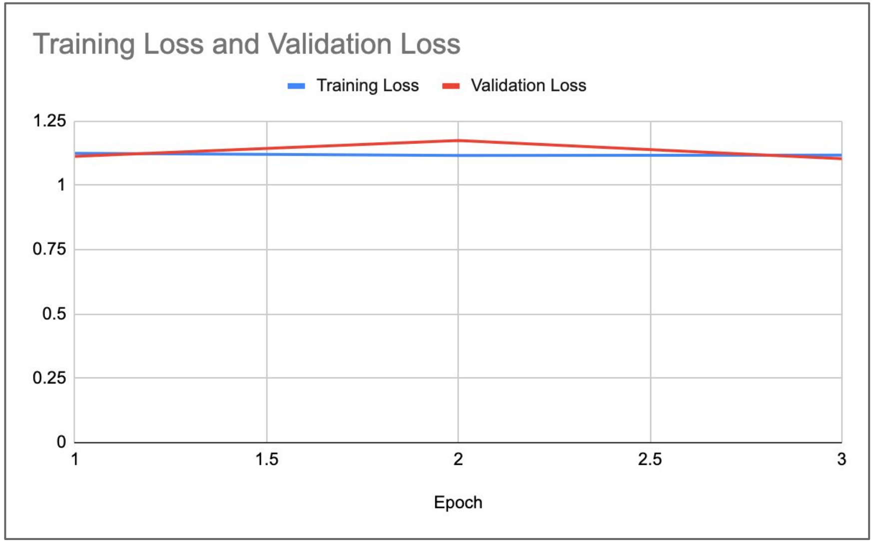
What’s going on here?

Digital Reasoning

Fiddling with Learning Rate (original slide)



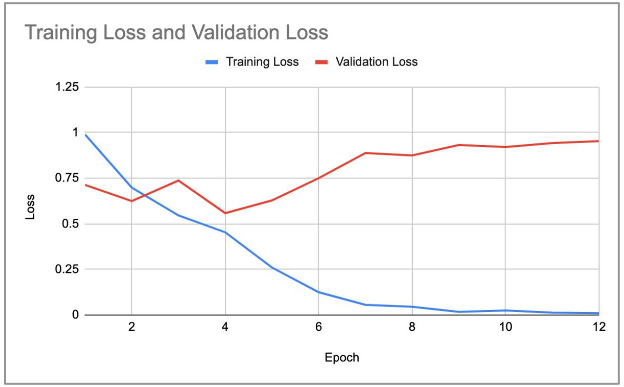
LR / 10



LR * 10

Digital Reasoning

Fiddling with Learning Rate (updated slide)



Learning Rate / 20

We'll get there eventually!

Default Learning Rate

Already overfitting after epoch #4

Learning Rate * 20

We'll never get there!

Digital Reasoning

The Colab notebook for this example is in the repo. Look at the final scores for the default case. Re-run that example with only 4 epochs and compare results.

Interesting/Relevant Links

Fast.ai: https://course.fast.ai/

Deep Learning for NLP and Speech Recognition. Kamath, Liu, and Whitaker. https://www.amazon.com/Deep-Learning-NLP-Speech-Recognition/dp/3030145956

BERT Explanation (very thorough): https://yashuseth.blog/2019/06/12/bert-explained-faqs-understand-bert-working/

Tweet Stance via Transfer Learning: https://towardsdatascience.com/transfer-learning-in-nlp-for-tweet-stance-classification-8ab014da8dde

The Illustrated Transformer: http://jalammar.github.io/illustrated-transformer/

Setting the learning rate of your neural network: https://www.jeremyjordan.me/nn-learning-rate/

Tuning BERT with the Fast.ai library: https://forums.fast.ai/t/a-tutorial-for-fine-tuning-bert-in-fastai/46569

GoogleBERT Code with Cloud TPU: https://colab.research.google.com/github/tensorflow/tpu/blob/master/tools/colab/bert_finetuning_with_cloud_tpus.ipynb#scrollTo=tYkaAlJNfhul

Sebastian Ruder’s Blog (also get his excellent newsletter): http://ruder.io/

NLP Progress Tracker: https://nlpprogress.com/

Digital Reasoning

Thank you

Contact me: joseph.porter@digitalreasoning.com