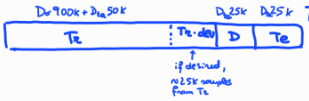
Deep Learning Specialization

MAIN CONCEPTS
- BACKPROPAGATION
- BIAS/VARIANCE ANALYSIS
- train/(train.dev)/dev/test split : D_{ta} = data similar to final task (100k)


D_{ot} = slightly different data (900k)

1. Neural Networks and Deep Learning

W1 - Structured vs unstructured data

W2 - Logistic regression : loss function, cost function, gradient descent
- Computational graph : foward propogation, backpropogation
- Gradient descent over many samples : solving logistic regression, vectorization.

W3 - My summary about backpropogation (W2-W3)
- Gradient descent for a 2-layer network : intuition and implementation for an L-layer network.
- Activation functions
- Weights initialization (basics)

W4 - Gradient descent for an L-layer network (summarized in W3)
- Why deep NN ?
- Caching in gradient descent.

2. Improving Deep Neural Networks

W1 NN DESIGNS TO REDUCE BIAS AND VARIANCE
- Train/dev/test split
- Bias and variance : intuition, metrics, no tradeoff (orthogonality)
- Reducing bias : depth, training time, architecture
- Reducing variance : Regularization (weight decay), "inverted" dropout, data augmentation, early stopping

AVOIDING CONVERGENCE PROBLEMS
- Speeding up training : data standardization.
- Batch normalization (end of 2.W3)
- Vanishing/exploding gradient : intuition of cases, advanced weight initialization (He)
- Gradient checking

W2 OPTIMIZERS
- Minibatch GD
- Minibatch size : Stochastic/Minibatch/Batch GD
- Smoothing sequences : Exponentially weighted average, bias correction
- Alternative optimizers : - GD with momentum (smoothed gradient)
  - RMSprop (adaptive scaling of the partial derivatives)
  - ADAM (adaptive momentum)
- Learning rate decay
- NOTE : few local minima and many saddle points in high-dim spaces.

W3 HYPERPARAMETERS OPTIMIZATION
- Relevant hyperparameters : xxx learning rate
  xx momentum, minibatch size, # hidden units
  x learning rate decay, # layers

Also addressed in course 3, in terms of
1. reducing bias (NN convergence, training set performance)
2. reducing variance (test set performance)

- Hyperparam search : - use random search
  - coarse to fine strategy
  - sample real hyperpars from log-uniform probability
  - tuning strategies : babysitting one NN Vs parallel training many NN

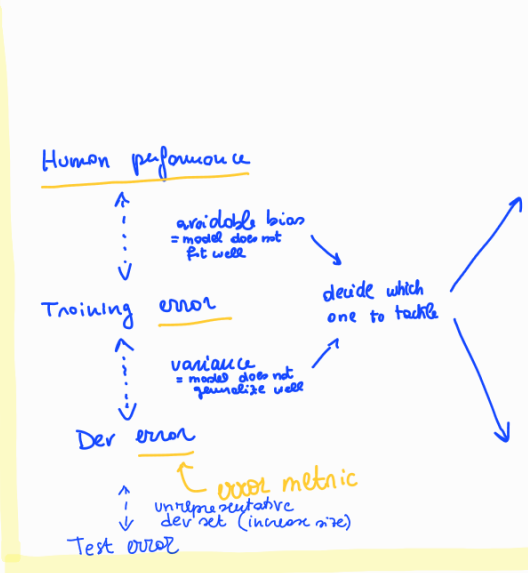
MISC
- Batch normalization ("Batch norm") : extends W1 on improving model convergence and reducing variance
- Softmax regression
- Intro to Tensorflow

3. Structuring Machine Learning Projects

W1 A GOOD ERROR METRIC IS THE BASIS FOR BIAS/VARIANCE ANALYSIS

- dev set and (unique) performance evaluation metric: define immediately, adjust as you go
- distinguish between metrics to be OPTIMIZED vs metrics to be satisfied
- use human-level performance (Proxy for Bayes optimal error) to determine avoidable bias.

"ORTHOGONAL" IMPROVEMENT OF ML MODEL (putting together course 2 and 3)



Model training:
- Smart weights init (avoid exploding/vanishing gradient)
- Train longer = # epochs
- Optimizer type
- Learning hyperparams = step size, minibatch size, optimizer's hyperparam, learning rate decay

Model architecture: - Batch normalization, # layers, # hidden units, activation functions

Debug problems: - Gradient checking

Model regularization: L2, dropout, (early stopping)

Expand training set: different TR/DEV/TE split, data augmentation

W2 MANUAL ERROR ANALYSIS

- motivation: integrates bias/variance analysis (identify causes of error and guides next steps).
- intuition

DATA MISMATCH (training and testing on different distributions)

- only between train and dev/test
- bug vs feature (when to use it as a feature)
- if feature: train / train-dev / dev / test splits
- if bug: synthetic data to reduce data mismatch

TRANSFER LEARNING

- used when not enough data for a task (like for data mismatch as feature) but pretrained net on similar task is available (unlike data mismatch as feature).
- idea

MULTITASK LEARNING

- used when not enough data for several similar tasks (mutual boosting)
- idea

END-TO-END LEARNING

- idea
- e2e learning vs intermediate problem learning depending on data availability.

4. Convolutional Neural Networks

W1 CONVOLUTION OPERATION

- basic principle & convolution over volumes
- padding & stride

ADVANTAGES OF CONVOLUTION (for images)
- convolutional filters capture spatially local features
- parameter sharing: same filter captures feature in different locations

CONVOLUTIONAL LAYER
- bias & activation
- no. hyperparameters

MAX POOLING LAYER

CNN
- Conv1 - pool1 - ... - convn - pooln - FC1 - FC2 - softmax
- through the network activations shrink but become deeper

W2 STANDARD CNN ARCHITECTURES

LeNet-5 : basic conv - pool - conv - pool - FC - FC - softmax

AlexNet : same as lenet but with more layers

VGG : much more regular than previous ones (only conv3,3, pool2,2, depth increase by a factor 2), many layers (depth is convenient).

RESNET: Creates very deep network avoiding the problem of potential redundancy across multiple layers. Ideally, if there are redundant layers one wants them to learn an identity mapping, which is however very hard in practice. By adding skip connections, redundant layers are fine with learning a zero mapping instead.

1x1 CONVOLUTIONS: are a way to compress channel information

INCEPTION: instead of choosing which filter to apply (3x3? 5x5? etc), define an inception module where many filters are applied in parallel and then compressed across channels using a conv1 to avoid explosion of channel no. In practice the inception module does conv1 ("bottleneck layer") followed by multiple filters in parallel. Reducing the no. channels with conv1 before applying filters considerably reduces computation.

NOT ENOUGH DATA...
- Transfer learning
- Data augmentation

W3 OBJECT LOCALIZATION AND DETECTION

Object or landmark localization

Evaluation of object localization using intersection over union

Object detection : - with sliding window
- convolutional implementation (last layer not flat)
- YOLO : convolutional implementation +
localization (bounding box prediction) in grid cells +
multiple anchor boxes per cell +
non-max suppression of overlapping BBs
- Object detection with region proposal (R-CNN, fast R-CNN, faster R-CNN).

Semantic Segmentation : Using U-Net architecture
- intuition : extract high-level semantic information (CNN) but
retain spatial information (transpose convolution + skip connections).

W4 FACE RECOGNITION

Learning similarity function
- Siamese Network + Triplet loss
- Siamese Network + logistic head

NEURAL STYLE TRANSFER
- Idea : given content and style images, iteratively refine random generated image.
- Use pretrained CNN as feature extractor, compute cost function (content & style),
freeze weights and backpropagate to change G.

5. Sequence models

W1

RECURRENT UNITS (RU)
- compact and unraveled representation

RNN ARCHITECTURES: one/sequence to one/sequence

LANGUAGE MODELLING TASK (train: sequence to one, test: one to sequence)
- tokenization & mapping to vocabulary
- word probability estimation = softmax over vocabulary
- training (cross-entropy loss) and inference

GRU & LSTM units

BIDIRECTIONAL RNN

DEEP RNNs

W2

ONE-HOT ENCODING OF WORDS

WORD EMBEDDING
- Property: analogy
- using word embedding = transfer learning
- Embedding matrix

WORD2VEC EMBEDDING:
- embedding of a word based on its local context.
- learning a skip-gram model (dataset = skip-gram pairs $< \overset{x}{\text{context}}, \overset{y}{\text{target}} >$) with standard Softmax
- learning a negative-sampling model (version of skip-gram) via logistic regression

GLOVE EMBEDDING:
- embedding of a word depends on its global co-occurrence in the entire corpus.

DEBIASING AN EMBEDDING:
- identify bias direction, neutralize bias in not definitional words, equalize bias.

W3

SEQUENCE TO SEQUENCE MODELS
- Encoder-decoder architecture
- Training: like language modelling
- Inference: . Beam search to find most likely output sequence
  . Beam search with length normalization
  . Error analysis on beam search

ATTENTION MECHANISM (pre-transformers)
- Review

W4

TRANSFORMERS (REVIEW)
- Self-attention
- Multi-head attention
- Overall architecture and refinements

$$u - v + w$$

$$a \cdot b - a - c + b * c$$