Study Guide for Deep Learning Specialization by Andrew Ng

Shuhao Lai

7/19/19

*Expect to review notebook for more thorough tips and applications.

Neural Network and Deep Learning

Summary

- Loss function is for one training image.
- Cost function is the average of the loss function for the entire training set.
- Avoid for-loops; use libraries instead.
- Broadcasting in Python copies elements along one axis when necessary.
- Know formulas for forward propagation per layer for set of images.
- Know tanh function's graph, equation, and derivative.
- Know ReLU's graph, equation, and derivative.
- Must use for-loop to iterate layers.
- Know backpropagation formulas for one training example.
- Know backpropagation formulas for training set.

Improving Deep Neural Networks

Week 1

- Know L2 regularization.
- Know dropout method.
- Know early stopping.
- Know what it means to normalize the training set.
    - Normalize data set by using mean and variance. The centers the data and creates even spread.
    - Divide data by variance to set variance to 1.
- Know weight initialization in EACH LAYER to help shrinking/exploding gradient.
    - Initialize weights to reduce shrinking and exploding gradient with the formula $w^{[i]} = np.random.randn(shape) * np.sqrt(\frac{x}{n^{[i-1]}})$
        - Np.random.randn produces an array filled with random floats sampled from a normal distribution of mean 0 and variance 1.
        - This formula changes the variance. Changing the variance works to reduce the weight as more of them are multiplied together.
- Know how to apply gradient checking.
- Know mini-batch gradient descent.
    - If batch is too small, you lose speed of vectorization.
    - Choose a size that is a power of two; $2^{[6-9]}$ are common choices.

- Know exponentially weighted averages.
  - Know what its name means.
  - Know what bias correction is.
- Know gradient descent with momentum.

Week 2

- Know RMSprop.
- Know Adam optimization algorithm.
  - Memorize this rather than RMS and momentum.
- Know three formulas for learning rate decay.
- Know that local minimum is not a big concern; plateaus are bigger concerns.

Week 3

- Hyperparameters and their importance (Left is the most important):
  - Learning rate; mini batch size, # hidden units, β; # layers; $\beta_1, \beta_2, \varepsilon$
- Know basic hyperparameter search for random sampling.
- Know how to perform log-base random sampling.
  - Using a log scale allows sampling more thoroughly at wiser magnitudes.
- Know how to perform Batch Normalization.
  - Used to allow each layer to expect a standard format for inputs.
- Know softmax layer and its cost function.

Structuring Machine Learning Projects

Week 1

- Dev and test set must come from the same source.
- Make set test big enough to give you confidence on performance; this could be 1% or 20% if total data.
- Know Bayer’s error.
- Know how to measure avoidable bias and variance.
- Human error ≈ Bayer’s error.
- Ways to reduce avoidable bias:
  - Train bigger model.
  - Train longer/better optimization algorithms.
    - Momentum, RMSprop, Adam
  - NN architecture/hyperparameters search.
- Ways to reduce variance:
  - More data.
  - Regularization.

        - L2, dropout, data augmentation.
- Know difference between train, dev, and test set.

Week 2

- Know how to perform error analysis to see which direction to go for changes.
- Know how to measure avoidable bias, variance, and data mismatch when train and dev/test data are from different sources.
- KNOW TRANSFER LEARNING.
    - Useful when task A&B have the same input, there is a lot more data for task A than task B, and low-level features form A could help for learning B.
- It is okay for training examples to be randomly and partially labelled wrong.
- Know what multitask learning is and when to use it:
    - Practically classifying more than one thing using one NN.

Convolutional Neural Networks

Week 1

- Large parameters more easily lead to overfit.
- Paddings helps retain dimension and allow edge pixels to be represented more frequently.
- Each filter needs to match the number of channels in input. Also, each shift in filter produces one value.
- Multiple filters produce multiple output layers.
- The size and shift of pooling and filters are hyperparameters.
- Max pooling > avg pooling.
    - Pooling layers usually have no padding.
    - Filter for it is usually 2x2 with shift of 2.
    - No parameters to learn.
- Hyperparameters are tunes while parameters are learnt.
- Between layers, activation size gradually, not sharply, decreases.
- CNNs have translation invariance.
- Models use in research often work well for a similar task. Know:
    - LeNet-5
    - AlexNet
    - VGG-16
- Know how residual blocks function and work.
- Know how residual blocks can work for CNNs.

Week 2

- Know 1x1 convolutions:

  - Useful to adjust number of channels while maintaining input's height and width.
  - Useful to reduce computation cost.
- Know inception network; allows multiple filter sizes and pooling in once layers.
- Know inception module used in research paper.
- Be on the look out for open source code for your own projects. TRANSFER LEARNING can be used.
- Transfer Learning Tips:
  - When transfer learning, might be worthwhile to compute and store the activation for layers not being trained.
  - If you have a large amount of your own data, consider freezing only a small portion of the prebuilt network; train on the last several layers.
  - Could retrain entire network.
  - VERY USEFUL AND WHAT YOU SHOULD ALMOST ALWAYS DO.
- Data augmentation techniques:
  - **Flip horizontally, random cropping,** shearing, rotation, local warping, **color shifting**.
- For benchmarking and competitions:
  - Ensembling: Train several networks independently and average their outputs.
  - Multi-crop at test time: run classifier on multiple versions of the test images and average results.
    - 10-crop is widely used.

Week 3

- Know the difference between image classification, classification with localization, and detection.
- Classification with localization has out vector = $\begin{bmatrix} p_c \\ b_x \\ b_y \\ b_h \\ b_w \\ c_1 \\ c_2 \\ c_3 \end{bmatrix}$ and can have a cost function like the following: $L(\hat{y}, y) = \{ \begin{matrix} (\hat{y}_1 - y_1)^2 + (\hat{y}_2 - y_2)^2 + \cdots + (\hat{y}_8 - y_8)^2, if\ y_1 = 1 \\ (\hat{y}_1 - y_1)^2, if\ y_1 = 0 \end{matrix}$
- Remember that backpropagation minimizes loss function not cost function.
- Know what landmark detection is.
- Know expensive method for object detection via sliding window.
- Know convolution implementation of sliding windows.
- KNOW YOLO ALGORITHM
- Know IOU.
- Know non-max suppression.
- Know Anchor boxes.
  - Not that likely to have mid-points within the same cell.
- Know region proposal object detection method:

  - Used because sliding window classifies too many regions where no objects exists.
  - Know R-CNN.
  - Know fast R-CNN.
  - Know faster R-CNN.

Week 4

- Verification:
  - Given input image and name/ID.
  - Outputs whether the input image is that of the claimed person.
- Recognition:
  - Has database of k persons.
  - Given input image.
  - Output ID if the image is any of the k persons (or not "recognized")
- Know what the one-shot learning problem is and basic idea to solve it.
- Know siamese network.
- Know triplet loss.
  - Know anchor, positive, negative.
  - Know loss and cost function.
  - Choose triplets that are "hard" to train.
- Some of pretrained networks are available online!
- Know face verification and binary classification, which is an alternative to triplet loss, to solve one-shot learning problem.
  - Know the two equations to compute output.
  - Save f(x) for images in database and the two networks are tied siamese networks.
  - Training involves pairs of images and labels of whether they are different are the same.
- Know how to apply backpropagation to siamese networks.
- Know intuition of what is happening at each layer for CNNs.
- Know neural style transfer cost function:
  - $J(G) = \propto J_{content}(C, G) + \beta J_{style}(S, G)$
- Gradient descent changes image, not weights (assuming you used a pretrained network)
  - $G_{new} = G_{old} - \frac{\partial}{\partial G} J(G)$
- Know content cost function.
- Know style cost function.
- Know that CNN can be applied to 1D, 2D, and 3D data.

Sequence Models

Week 1

- Know vocabulary vector and one-hot vector for each word.

- RNN better than CNN or fully connected because:
    - Input and outputs can be different lengths.
    - RNN can share featured learned across different position of texts.
- Know forward propagation for all RNN architectures.
- Know how to perform backpropagation through time:
    - Know loss and cost function.
- Know the following RNN architectures:
    - One to one
    - One to many
    - Many to one
    - Many to many ($T_x = T_y$)
    - Many to many ($T_x \neq T_y$)
- Know what language modelling does.
- Know <UNK> and <EOS> tags.
- Know how to sample from language model to generate novel output.
- Know how to deal with exploding gradient (clipping)
- Vanishing gradient:
    - Know Gated Recurrent Unit.
    - Know Long Short Term Memory (More powerful than GRU)
- Know bidirectional RNN
- Know deep RNNs
    - Not that deep, ~3 layers

Week 2

- Know what word embedding is.
- Can download pretrained word embeddings online to use for your own task.
- Know how to use embeddings for analogies.
    - Use cosine similarity.
- Know the form of embedding matrix.
- Training word embeddings:
    - Know neural language model using context and target words to train embeddings (word2vec). This is the first, most complicated form. Sub methods are derivatives.
        - Know skip-gram (a type of word2vec) and its problems.
            - Know hierarchical SoftMax
            - Sampling context word is difficult.
        - Know negative sampling.
            - Know how to choose samples to avoid choosing “the” frequently.
    - Know GloVe method. Not the same as word2vec.
        - Confusing, must review.
- Sentient classification reads a sentence and identifies whether the person likes or dislikes the topic.
    - One method is to average embeddings.

        - Doesn’t consider word order.
    - Another method is to use a many to one RNN.
- Know what debiasing word embeddings means:
    - Know how to perform each step to get rid of bias:
        - Identify bias direction.
        - Neutralize bias in non-definitional words.
        - Equalize pairs of words that are defined by bias, like gender.

Week 3

- Know the basic encoding and decoding translation model.
    - Machine translation is the same as building a conditional language model.
- Know how to perform image captioning.
- Greedy search for machine translation will not produce the best translation because you want to consider the entire output together that maximizes the probability and this is not guaranteed to occur picking one word at a time.
- Know how to perform beam search.
- The following refinement to beam search prevents numerical overflow and removes the functions to unnaturally favor shorter sentences.
    - $\frac{1}{T_y^{\propto}}\sum_{t=1}^{T_y} \log p(y^{<t>} | x, y^{<1>}, \ldots, y^{<t-1>})\, , where\ \ \propto = 0.7\ is\ effective.$
- Beam width choices:
    - B = 10 for production system.
    - B = 100 large for production system.
    - 1000-3000 research to squeeze out performance.
- Know Error analysis on beam search as used for translation to track down whether problem is with beam width or model.
- Know how to calculate Bleu Score:
    - $(brevity\_penalty)\exp\left(\frac{1}{4}\sum_{n=1}^{4} \frac{\sum_{n-grams\ \epsilon\ \hat{y}} count\_clip(n\_grams)}{\sum_{n-grams\ \epsilon\ \hat{y}} count(n-gram)}\right)$
    - Know how to calculate brevity_penalty.
- **KNOW ATTENTION MODEL**
- Preprocess audio clips with spectrograph to get y = frequency, x = time, and color = energy/loudness.
- Could process speech with attention model and or any other model.
- Know connectionist temporal classification (CTC) cost for speech recognition.
- Know how to implement trigger word system in RNN.