Workshop 07: Statistical Learning II: A Practical Course in Deep Learning for Statisticians: Neural Network Fundamentals with Healthcare Applications Using Tensor Flow

Evan Paul Carey
Saint Louis University

https://github.com/evan-paul-carey/deep-learning-workshop

1

1

Course outline

- Overview of deep learning
- Introduction to deep learning frameworks
- Regression as a NN
- NN fundamentals
- Applied universal function approximation
- Regularization
- Applied regression problem
- Recurrent NN and Convolutional NN extensions
- Medical extensions

2

2

Who are you and what is deep learning?

3

3

Instructor: Evan Carey

- Assistant professor of Health Data Science @ Saint Louis University
- Epidemiologist/Biostatistician/Health data scientist @ Department of Veteran Health Affairs
- MS Applied biostatistics, PhD Epidemiology
- Interest in answering useful questions using big healthcare data.

https://github.com/evan-paul-carey/deep-learning-workshop

Evan.carey@va.gov

Evan.carey@health.slu.edu

4

4

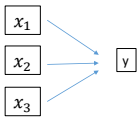
Attendee poll

- Raise your hand if you have had coursework/training in general linear models with p-values ☺
- Raise your hand if you have had coursework/training in machine learning approaches to generating predictive algorithms ☺
- Raise your hand if you have had coursework/training in deep learning models with Tensorflow ☺

5

5

Statistical learning



- Traditional inferential statistics / Bayesian inference
  - Goal: approximate the data generating process while make inference on uncertainty.
- Machine learning
  - Goal: create an algorithm that predicts y with the highest accuracy.

6

6

Deep learning as a subset of ML

- Deep learning is a subset of the field of ML.
- Similar idea/goal:
  - Transform inputs into meaningful representations (outputs)
  - Given an image, transform it into a prediction of 'Cat or not cat'
  - Given demographics, medical comorbidities, and recent clinical utilization, predict total cost of care next year.
  - Given demographics, medical comorbidities, and recent clinical utilization, predict probability of mortality in the next 6 months.

7

7

Deep learning versus 'other ML' approaches

- One fundamental application difference is the degree of feature engineering required.
- Under deep learning frameworks, we can flexibly add inputs. Since we model 'hidden units' (more on this later), we can hopefully extract meaningful feature engineering directly from the labelled data.
  - Why didn't we do this prior to the current deep learning craze?

8

8

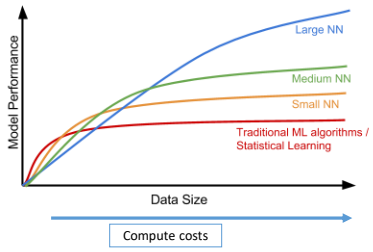
Why deep learning now?

- Deep learning has become exceptionally popular in the last 10 years.
- Why do you think that is?

9

9

The potential of deep learning



Source: adapted from Andrew Ng's course videos.

10

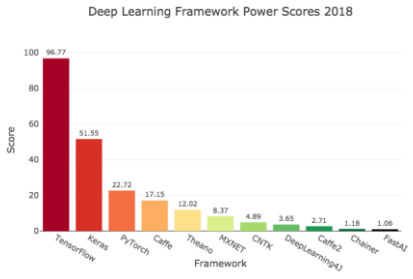
10

Popular deep learning frameworks

- Tensorflow
  - Associated with Google
- Pytorch
  - Younger than Tensorflow (1.0 released October 2018)
  - Growing ***quickly*** in popularity
  - Backed by Facebook
- Microsoft Cognitive Toolkit (CNTK)
  - Microsoft solution
- Apache MXNet
  - Used by Amazon

11

11



https://towardsdatascience.com/deep-learning-framework-power-scores-2018-23607ddf297a

12

12

Keras

"Keras is a high-level neural networks API, written in Python and capable of running on top of TensorFlow, CNTK, or Theano. It was developed with a focus on enabling fast experimentation. *Being able to go from idea to result with the least possible delay is key to doing good research.*"
- Keras documentation

- Works on CPU and GPU
- Allows you to easily specify models
- Supports CNN and RNN
- Works with Python 2.7-3.6
- User friendly API for deep learning.

13

13

TensorFlow + Keras

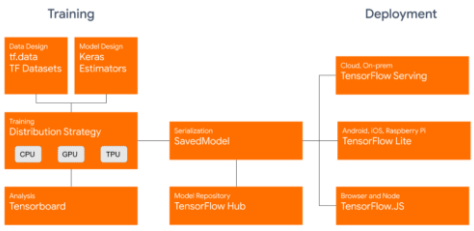
- Tensorflow has adopted and integrated a version of Keras directly into TF!
  - Tensorflow.keras =/= keras
- Although keras was not originally tied to Tensorflow, future development will be tied to tensorflow. More info here:

https://www.pyimagesearch.com/2019/10/21/keras-vs-tf-keras-whats-the-difference-in-tensorflow-2-0/

14

14

Tensorflow 2.0 available as of Sept 30, 2019 (oh my.)



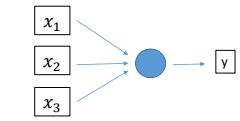
https://blog.tensorflow.org/2019/09/tensorflow-20-is-now-available.html

15

15

Linear regression as a form of (shallow) deep learning.

$$y_i = \beta_0 1 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + e$$



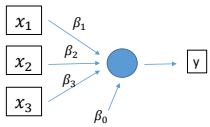
Input layer Hidden layer Output

16

16

Construct a graph for this feed forward NN:

- For every hidden layer:
  - Add a weight for each previous connected layer, plus one an offset weight.
  - Sum these weights (betas) times inputs plus offset (affine function)
  - Output some function of this sum.



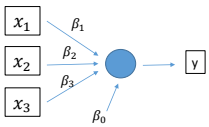
$$y = \beta_0 1 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3$$

Input layer Hidden layer Output

17

17

Logistic regression as a form of deep learning.



Single hidden layer – identity link, sum of weights (betas) times inputs plus offset

Output layer – logit function (sigmoid function)

18

18

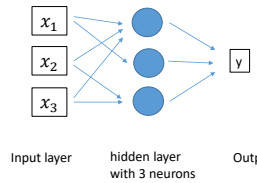
Loss functions and optimization

- What is the loss function for linear regression?
  - Mean squared error (least squares regression).
  - What about other loss functions?
    - Mean absolute error?
- What is the optimization algorithm/approach for typical linear regression (like `lm()` in R)?

19

19

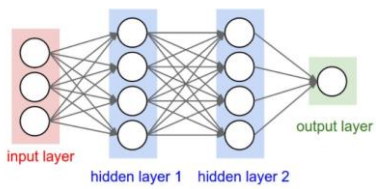
Linear regression as a form of deep learning?



20

20

Example feed forward artificial neural network (ANN)

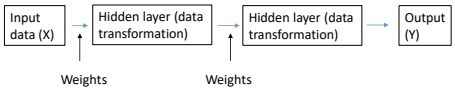


A three-layer artificial neural network.
(Image source: http://cs231n.github.io/convolutional-networks/#conv)

21

21

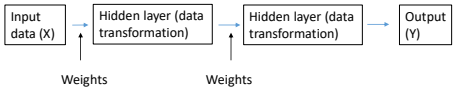
Deep learning goal – estimate the weights!



- Given a model architecture and observed data, estimate the weights.
- How do we know which weights are best?
  - optimization
- How do we avoid overfitting the training data?
  - regularization

22

22



- How do I know how good my current value of weights are?
  - Flow input data forward through the network (forward propagation)
  - Calculate cost (loss) function value, maybe squared error?
- How do I know which direction to modify my weights to get a 'better' cost function?
  - Estimate each gradient (maybe the 2nd order derivative too)
  - Backpropagation is used to adjust the weights across this complex function space.

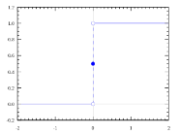
23

23

Activation functions more generally.

- In a NN, each neuron:
  - Receives values from the prior connected neurons, modified by weights and an offset
  - Linearly sums the inputs
  - Applies an activation function (potentially non-linear transformation of the sum of the inputs)
  - Classic neuron – On or Off (firing or not firing)
  - But what about other functions?

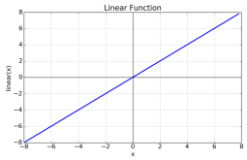
Step function:
- Value of 1 above threshold
- Value of 0 below threshold



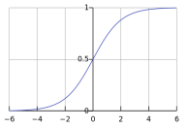
Graphs from - https://medium.com/the-theory-of-everything/understanding-activation-functions-in-neural-networks-9491262884e0

24

Linear function:
- Simple linear sum of inputs



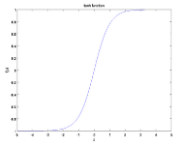
Sigmoid function:
- Constrains output to be between 0 and 1
- Logistic regression!



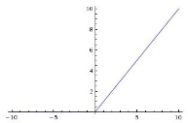
Graphs from - https://medium.com/the-theory-of-everything/understanding-activation-functions-in-neural-networks-9491262884e0

25

Tanh function:
- Scaled sigmoid



Rectified Linear Unit function (relu):
- 0 unless above threshold
- Linear thereafter



Graphs from - https://medium.com/the-theory-of-everything/understanding-activation-functions-in-neural-networks-9491262884e0

26

Using an ANN to approximate (almost) *any* function

- The universal approximation theorem tells us an ANN feed forward network with a finite number of neurons and a single hidden layer can approximate any continuous function.
  - What does finite mean ?
  - ReLU is often used these days for hidden layers and has been shown to perform well.

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

27

27

Choosing the ANN architecture

- When we design an ANN, we must design the architecture.
- In practice, this means specifying:
  - The number of hidden layers (the depth)
  - The number of units per hidden layer (the width)
  - The activation functions for each layer. ReLU is common and a good starting point!

28

28

Move to notebooks!

02_keras_deep_learning_intro.ipynb

29

29

Regularization for deep Learning

30

30

Why do we need to regularize our models (review)?

- Remember our goal…
  - maximize prediction on data we can see?
  - Or maximize prediction on data we can't see?
- Regularization hurts our training error.
  - But hopefully it improves our validation error!
- We are introducing a lot of flexibility to the model through a deep/wide neural network…
  - The best fitting model in practice is typically a flexible model, regularized appropriately.

> Regularization is "any modification we make to a learning algorithm that is intended to reduce its generalization error but not its training error."
> - Chapter 5, Deep Learning Book, Ian Goodfellow

31

31

How do I track generalization error?

- Generalization error is inherently unobservable.
- We need creative approaches to hold out samples (validation / test) to replicate the notion of 'out of sample data'.
  - Hold out across time units, or clusters…
- After constructing a validation set, we can use to validate hyperparameters.
  - We can also track our estimate of training and generalization error across epochs.

32

32

Learning Curves

Early stopping: terminate while validation set performance is better

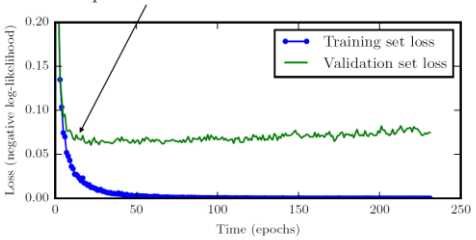


Figure 7.3

33 (Goodfellow 2016)

33

L1 / L2 Penalties – parameter norm penalties.

- Typically we penalize the weights but not the bias terms of the affine transformation for each neuron.
- We will typically use the same weight across the network, although we could consider optimizing a different weight for each neuron with input(s).
  - This would be expensive (large hyper-parameter space to search).
- L2 penalty
  - Like ridge regression. Reduces correlated inputs, but leaves weights in the model as small values.
- L1 penalty
  - Like LASSO regression. Eliminates some weights, induces sparsity.

34

34

L1/L2 regularizers in keras

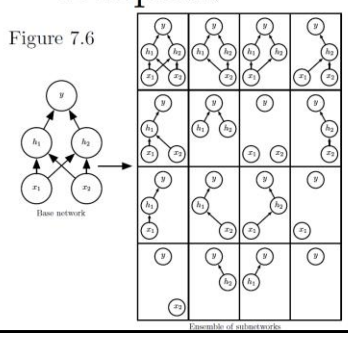
- kernel_regularizer:
  - This will regularize the weights matrix (the Beta's)
- activity_regularizer:
  - This will regularize the output of the neuron (the activation function output)
- bias_regularizer:
  - This will regularize the bias term for the neuron

35

35

Dropout

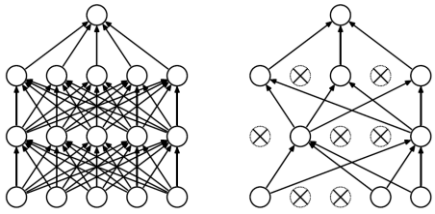
Figure 7.6



36

36

Dropout (figure from reference)



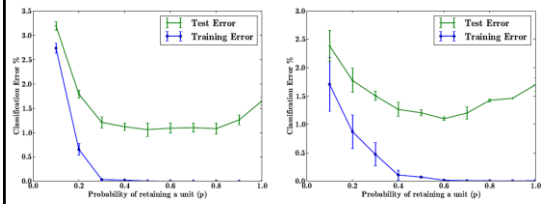
(a) Standard Neural Net (b) After applying dropout.

Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*. 2014;15(1):1929–1958.

37

37

How to pick the dropout probability?



(a) Keeping n fixed. (b) Keeping pn fixed.

Figure 9: Effect of changing dropout rates on MNIST.

Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*. 2014;15(1):1929–1958.

38

38

Boostrap aggregating (bagging)

- Also called 'bagging'
- Fit multiple different models on different resamples from the dataset.
  - Resample the dataset multiple times with replacement.
  - Fit a model on each resample.
  - If the different members make 'errors' in different ways, then the group vote will be stronger than any individual vote.
  - Deep learning is inherently stochastic in nature…random parameter initialization, random mini-batch selection…

39

39

Bagging

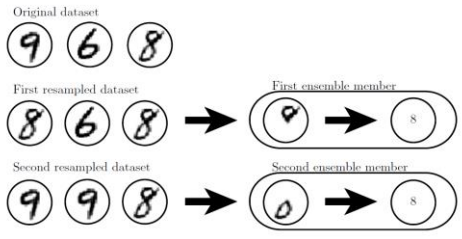


Figure 7.5

40 (Goodfellow 2016)

40

Move to notebooks ☺

03_deep_learning_regression.ipynb

41

41

Deep Learning Extensions: RNN and CNN

42

42

Feed-forward ANN review

- Do these networks have any 'memory' of past inputs or hidden layer outputs?

43

43

Modeling sequence data

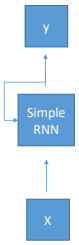
- Consider the following data points:
  - Age, race, gender
  - Comorbidities (potentially time-varying)
  - Longitudinal medication exposure (sequence data)
  - Emergency department visits
- Analytic goal –
  - Create a risk model that predicts emergency department utilization.

44

44

Recurrent Neural Networks

- RNN's are useful for processing sequences of data.
  - Feed forward struggle to use past values in additional to current values, unless we overtly include them as separate inputs.
  - The idea is that the RNN unit will 'remember' prior values (prior in time or more generally any index order)
  - A simple RNN –
  - In keras, this is the the 'SimpleRNN' layer

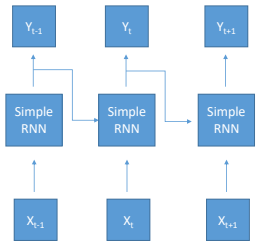


45

45

RNN Unrolled

- Each simple RNN is a function of current X, and the prior RNN output.
- In theory, all the information from prior X's could be included in the final output as we train the model.
- In practice, this doesn't work out.
  - Why?



46

46

Vanishing gradient issue with simple RNN.

- A simple RNN has a problem with vanishing gradients across earlier time-steps.
  - In theory the network could learn everything, but in practice it fails dues to vanishing gradients of something optimized repetitively…
  - Occasionally the gradients will explode instead of vanish, causing optimization issues.

47

47

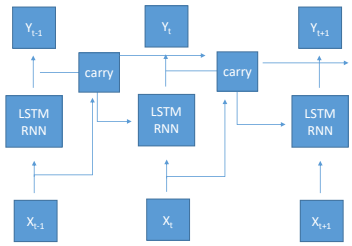
Long Short-Term Memory (LSTM)

- Developed by Hochreiter, Schmidhuber, and Bengio in 1997.
- Allows past information to be injected at future times, without being subjected to a vanishing gradient.
- Each LSTM RNN layer is a function of:
  - Current X input
  - Prior LSTM output (t-1)
  - 'Carry' information from prior time point

48

48

LSTM Diagram



49

49

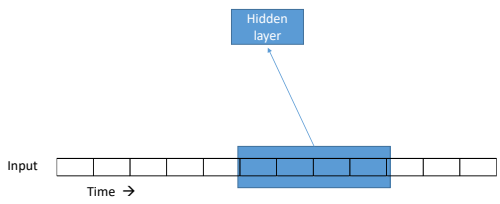
Other RNN layers available in keras

- Gated recurrent unit (GRU)
  - Published in 2014
  - Simpler / faster than an LSTM
  - Less capacity though…
- Stacking recurrent NN layers?
  - Each layer should return the full sequence (not just the last timestep value)
- Stacking multiple LSTM layers is *expensive*…

50

50

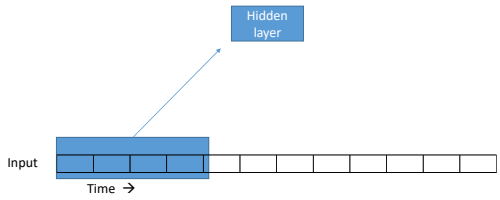
1D- Convolutions



51

51

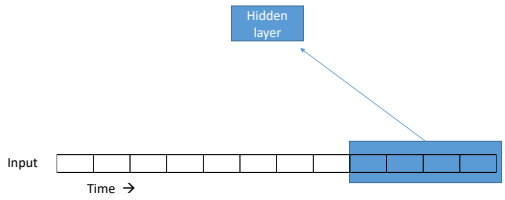
1D- Convolutions



52

52

1D- Convolutions



53

53

1D- Convolutions

- We must specify:
  - The window size (kernel_size in keras)
  - The number of convolutions (filters in keras)
  - Activation function
- These can be combined with recurrent layers like GRU
  - Use the 1d-Convolutions for dimensionality reduction
  - Use the GRU (or LSTM) to persist information across time points.

54

54

Deep learning review

- General deep learning approach:
  - Start simple and include a baseline (null information model)
  - Increase complexity and then regularize appropriately, checking for overfitting behavior graphically.
  - Continue to increase model capacity until you can no longer control overfitting via regularization
  - Keep best model.
- When and why to try deep learning?

55

55

Exciting Medical Applications

56

56

Estimating heterogenous treatment effects

- Estimating the conditional average treatment effect (CATE)
- This require a high capacity model (potentially at the expense of model clarity)
- Non DL approaches include:
  - Linear model based approaches (interactions)
  - Tree based methods
- But what about data structure complexity?
  - Time series within the data
  - Free text

Chen R, Liu H. Heterogeneous Treatment Effect Estimation through Deep Learning. *arXiv:181011010 [stat]*. October 2018. http://arxiv.org/abs/1810.11010. Accessed July 19, 2019.

57

57

Medical word embeddings

- Using clinical text from EMR's is challenging under traditional statistical learnings frameworks.
- NN can use text as inputs, then transform to lower dimensional embeddings for subsequent use in the model (NLP).
- Embeddings trained on big datasets can be made available, then incorporate into local (smaller) datasets for use. Weights can be updated with the smaller datasets (transfer learning!) .

Zhang Y, Chen Q, Yang Z, Lin H, Lu Z. BioWordVec, improving biomedical word embeddings with subword information and MeSH. *Scientific Data*. 2019;6(1):1-9. doi:10.1038/s41597-019-0055-0

58

58

Deep learning limitations

- Deep learning is not better for many (most?) problems utilizing clinical data.
- Deep learning is expensive and complex, often for no gain in predictive power.
- Deep learning may be harder to interpret than other approaches.
- Deep learning requires large amounts of labelled data, which is challenging in the context of observational medical data.

Chen D, Liu S, Kingsbury P, et al. Deep learning and alternative learning strategies for retrospective real-world clinical data. *npj Digital Medicine*. 2019;2(1):1-5. doi:10.1038/s41746-019-0122-0

59

59

More Tensorflow

- The tensorflow website has some great tutorials, with many more being developed as we speak. Check these out!

https://www.tensorflow.org/learn

60

60