A Brief Background on Deep Learning and the Building Blocks of Neural Networks

David Elsheimer & Peter Norwood

North Carolina State University

January 16th 2019

What is deep learning and how does it work?

- "Deep" refers to sequential layers of representations.
- The more layers a model has the greater the "depth".
- Layered representations generally learned through neural networks. (Not related to how the brain works)
- Loss function compares true and predicted responses. Loss scores fed into an optimizer to adjust weights in a given layer.

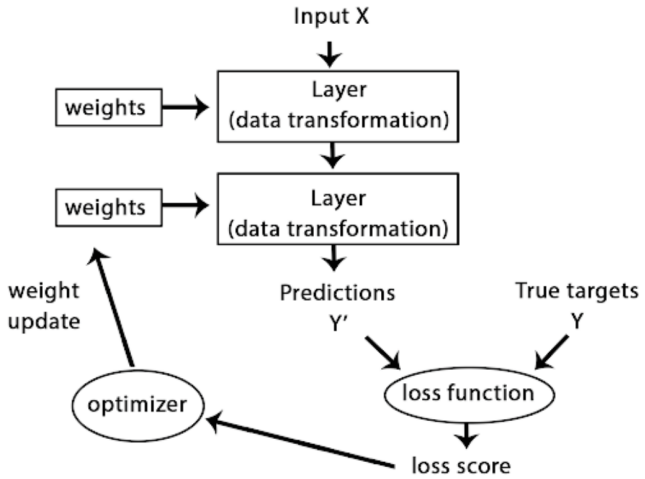

Figure 1.9 The loss score is used as a feedback signal to adjust the weights.

(Chollet & Allaire 8-11)

Deep Learning Benefits

While overly hyped, deep learning models will likely stand the test of time because they are:

- Simple to implement: Automated feature engineering, all features are updated when one feature is adjusted
- Scalable: Parallelize well
- Versatile and reuseable: Can be trained on new data without starting over; good for continuous learning (think instream data)

(Chollet & Allaire 22-23)

Data Representations & Tensor Operations

- Representations: Scalars (0D), Vectors (1D), Matrices (2D), "Data Cube" (3D), etc.
- Attributes: Rank (#axes/dimensions), shape (vector of tensor dimensions)
- Data types: integer, double, character, etc.
- Tensor operations: Element-wise operations to all entries, operations to tensors of different dimensions (MV multiply), tensor dot, reshaping
- Tensor dot example: 2×3 matrix dotted with 3×5 matrix results in a 2×5 matrix

(Chollet & Allaire 29-30, 35-38)

Gradient-based Optimization

- Gradient here is the derivative of a tensor operation (derivative of a function that takes tensors as inputs, output has same shape)
- Gradient step involves an intial tensor W_0, step akin to the form $W_1 = W_0 - \alpha \cdot \nabla f(W_0)$
- Analytically finding weights is intractable, uses minibatch stochastic gradient descent (performed on only part of the data)

(Chollet & Allaire 43-45)

Gradient-based Optimization

- Optimizer uses backpropogation (BP) algorithm which relies on applying chain rule (CR) to the computing of the gradient
- BP: Start at the last loss value, work from top to bottom using CR to compute each parameter's loss value contribution
- Newer frameworks (ex. TensorFlow) use symbolic differentiation; Given a chain of operations with a derivative, computes a gradient for the chain, backwards step reduces to gradient function call

(Chollet & Allaire 46-47)

References I

1 Chollet, François, and J. J. Allaire. Deep Learning with R. Manning Publications, 2018.