AlphaFold Bitesized Pieces

Reuben Brasher

November 19, 2021

How to be an ML Engineer

- Define cost function

How to be an ML Engineer

- Define cost function
- Define network architecture

How to be an ML Engineer

- Define cost function
- Define network architecture
- Apply gradient descent

Classical Neural net

Refer to Fig. 1.

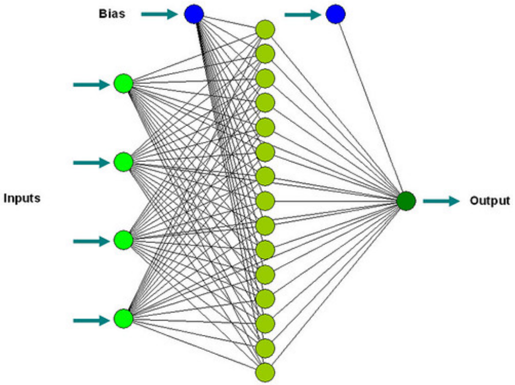

Figure: https://search.creativecommons.org/photos/70ab8654-c234-4dbe-9b1c-62851544245a

Dense Layers

- Linear function whose coefficients are parameters of model

$$y_j = \sum_i w_{ij} x_{ij} + b_j$$

Dense Layers

- Linear function whose coefficients are parameters of model

$$y_j = \sum_i w_{ij} x_{ij} + b_j$$

- Possible non-linear activation function

$$F(y)$$

or

$$f(y_j)$$

Common activation functions, tanh and sigmoid

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

$$\mathrm{sigmoid}(x) = \frac{1}{1 + e^{-x}}$$

Common activation functions, softmax

$$\text{softmax}(x)_j = \frac{e^{x_j}}{\sum_i e^{x_i}}$$

Common activation functions, relu

$$\mathrm{relu}(x) = \max(x, 0)$$

Gate Layers

- Entrywise multiplication of two previous layers outputs

$$(x \odot y)_i = x_i y_i$$

Gate Layers

- Entrywise multiplication of two previous layers outputs

$$(x \odot y)_i = x_i y_i$$

- Became popular with LSTM and GRU

Attention Layers

- Define a convex combination along axis of previous layer

$$y_i = F(x_i)$$

$$a_i = \text{softmax}(\text{linear}(y))_i$$

$$\sum_i a_i y_i$$

Attention Layers

- Define a convex combination along axis of previous layer

$$y_i = F(x_i)$$

$$a_i = \text{softmax}(\text{linear}(y))_i$$

$$\sum_i a_i y_i$$

- Became popular with question answering methods.

Transformers with Multi-head Attention Layers

- Multi-head attention. Layer produces three outputs q, k and v

$$\mathrm{softmax}(qk^T)v$$

Transformers with Multi-head Attention Layers

- Multi-head attention. Layer produces three outputs q, k and v

$$\mathrm{softmax}(qk^T)v$$

- Defined in Vaswani et al., 2017

Gradient descent

ϕ a real-valued function of net output $F(x)$ and possible labeled observation y
Θ the parameters (linear coefficients)
Cost is

$$\phi(F(x|\Theta), y)$$

Minimize with respect to parameters using gradient

$$\nabla_\Theta \phi(F(x|\Theta), y)$$

Encoder-decoder pattern

Train a pair of models, encoder to produce concise representation and decoder to reconstruct.
Later encoder and decoder can be used separately.

Bert: Pre-training of deep bidirectional transformers for language understanding

Devlin et al., 2018

Model pretrained to reconstruct corrupted text and then finetuned

Human pose estimation with iterative error feedback

Carreira et al., 2016

Self-training with noisy student improves imagenet classification

Xie et al., 2020

Deep residual learning for image recognition

He et al., 2016

References I

- Carreira, Joao et al. (2016). "Human pose estimation with iterative error feedback". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4733–4742.
- Devlin, Jacob et al. (2018). "Bert: Pre-training of deep bidirectional transformers for language understanding". In: *arXiv preprint arXiv:1810.04805*.
- He, Kaiming et al. (2016). "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.
- Vaswani, Ashish et al. (2017). "Attention is all you need". In: *Advances in neural information processing systems*, pp. 5998–6008.

References II

Xie, Qizhe et al. (2020). "Self-training with noisy student improves imagenet classification". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10687–10698.