Git Cheat Sheet

Modular Learning

Definition: A family of parametric, non-linear and hierarchical representation learning functions, which are massively optimized with stochastic gradient descent to encode domain knowledge, i.e. domain invariances, stationarity.

- Neural Network is a directed acyclic graph
- Use loss function that matches output distribution to improve numerical stability and make gradients larger
- Input and output distribution of every module should be the same to prevent inconsistent behavior and harder learning

Backprop: chain rule $\frac{\partial z}{\partial x_i} = \sum_j \frac{\partial z}{\partial y_j} \frac{\partial y_j}{\partial x_i}$, $\nabla_{\boldsymbol{x}} \boldsymbol{z} = \left(\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{x}} \right)^T \cdot \nabla_{\boldsymbol{y}} \boldsymbol{z}$

1. Compute forward: $a^{(l)} = h^{(l)} \left(x^{(l)} \right)$, $x^{(l+1)} = a^{(l)}$
2. Compute reverse: $\frac{\partial \mathcal{L}}{\partial a^{(l)}} = \left(\frac{\partial a^{(l+1)}}{\partial x^{(l+1)}} \right)^T \cdot \frac{\partial \mathcal{L}}{\partial a^{(l+1)}}$
   $\frac{\partial \mathcal{L}}{\partial \theta^{(l)}} = \frac{\partial a^{(l)}}{\partial x^{(l+1)}} \cdot \left(\frac{\partial \mathcal{L}}{\partial a^{(l)}} \right)^T$
3. Update params: $\theta_{t+1}^{(l)} = \theta_t^{(l)} - \eta \nabla_{\theta_t^{(l)}} \mathcal{L}$