Neural Network Function Approximation

Map of RL Algorithms

This Time

Challenges in Reinforcement Learning:

- Exploration vs Exploitation
- Credit Assignment
- Generalization

Function Approximation

input

$$y \approx f_\theta(x)$$

output parameters

Previously, Linear:

$$f_\theta(x) = \theta^\top \beta(x)$$

weights features

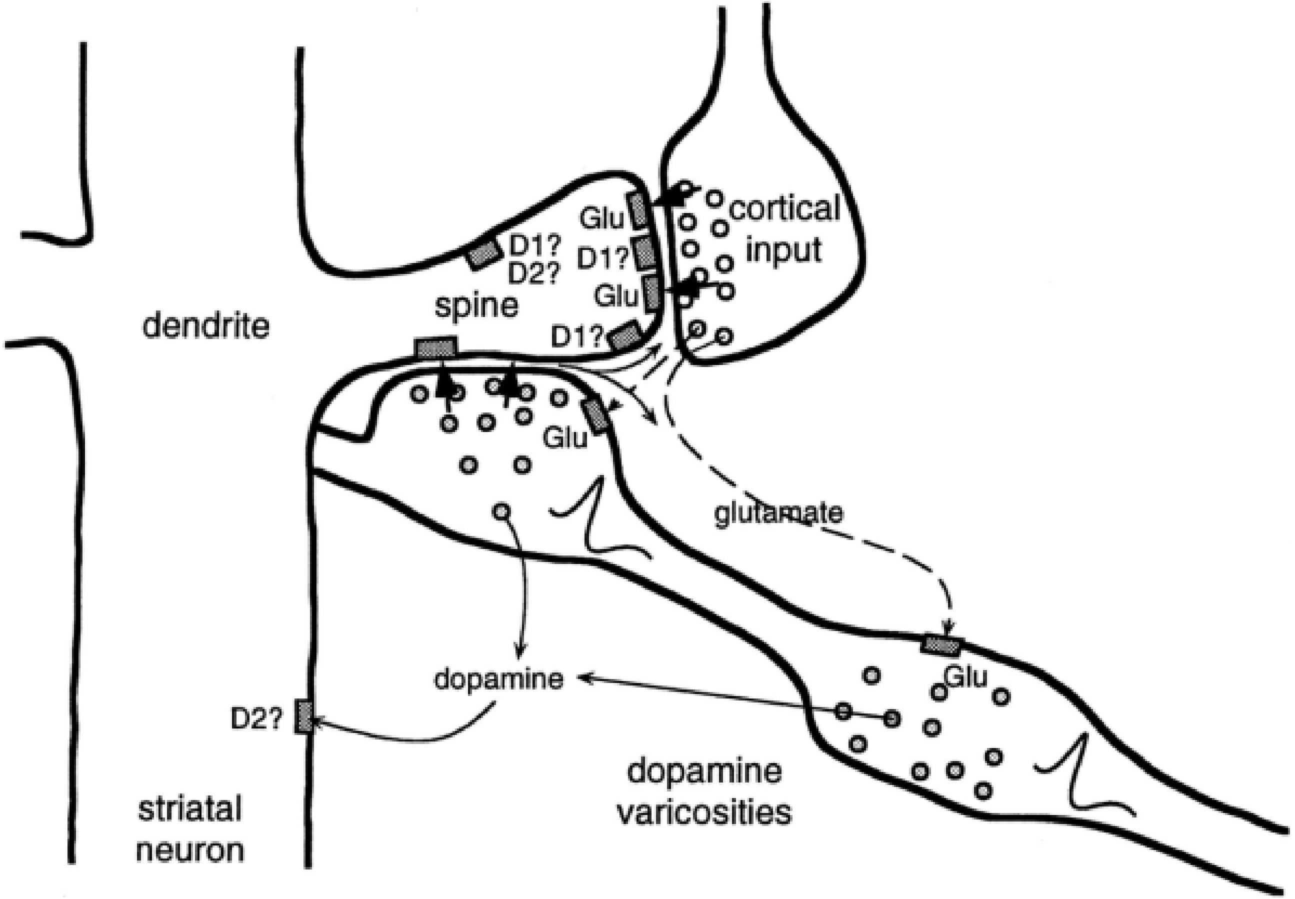
e.g. $\beta_i(x) = \sin(i\,\pi\,x)$

AI = Neural Nets

Neural Nets are just another function approximator

Neural Network

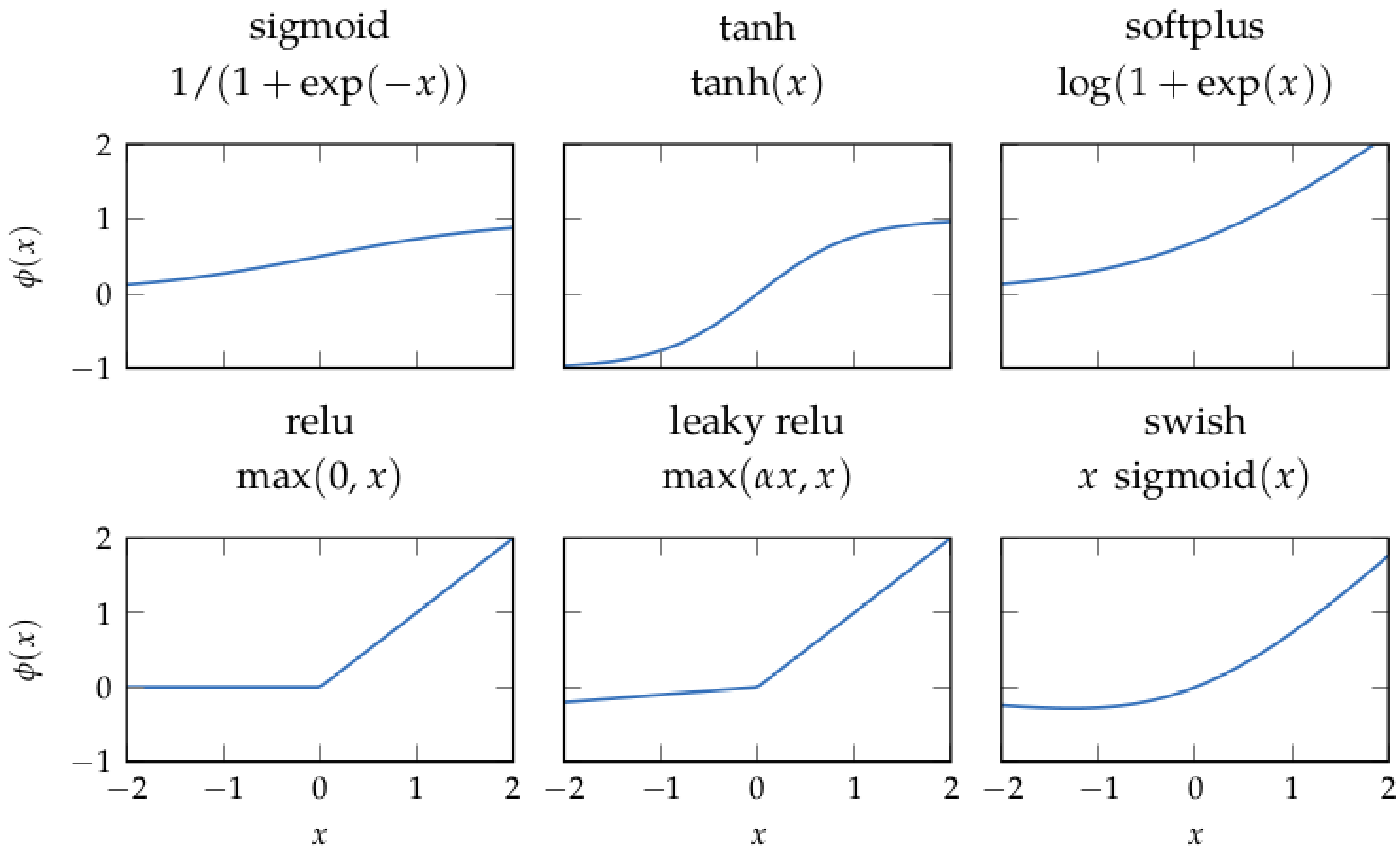
$$h(x) = \sigma(Wx + b)$$



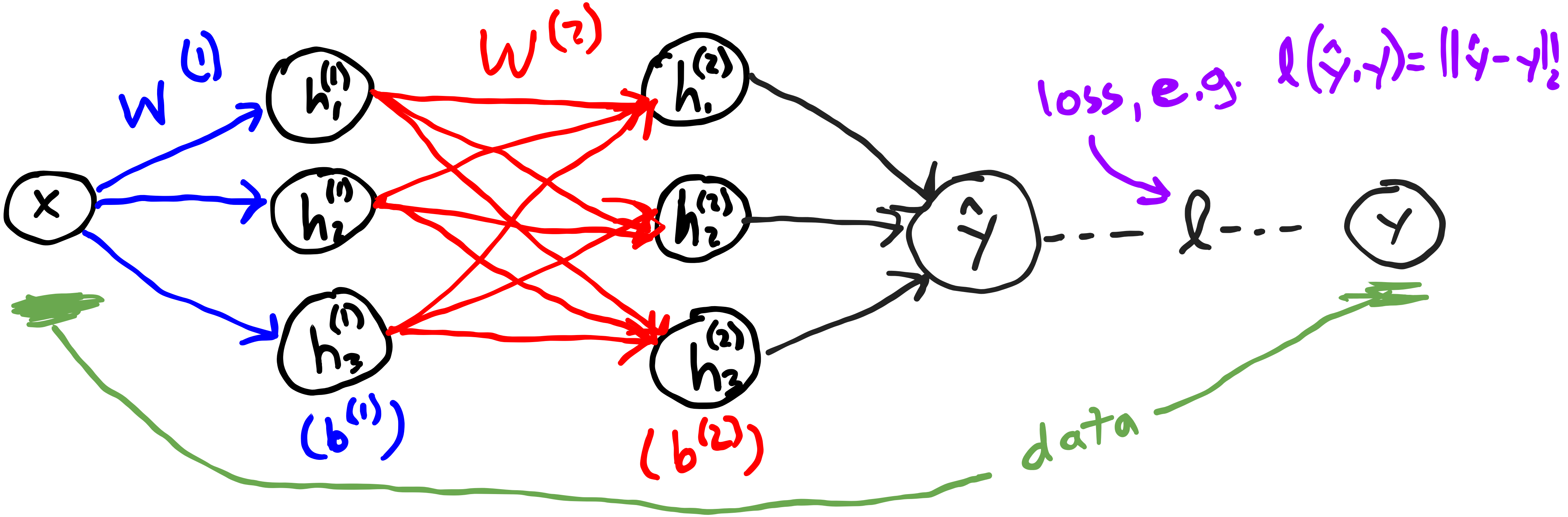
Neural Network

$$h(x) = \sigma(Wx + b)$$

Nonlinearities



Training

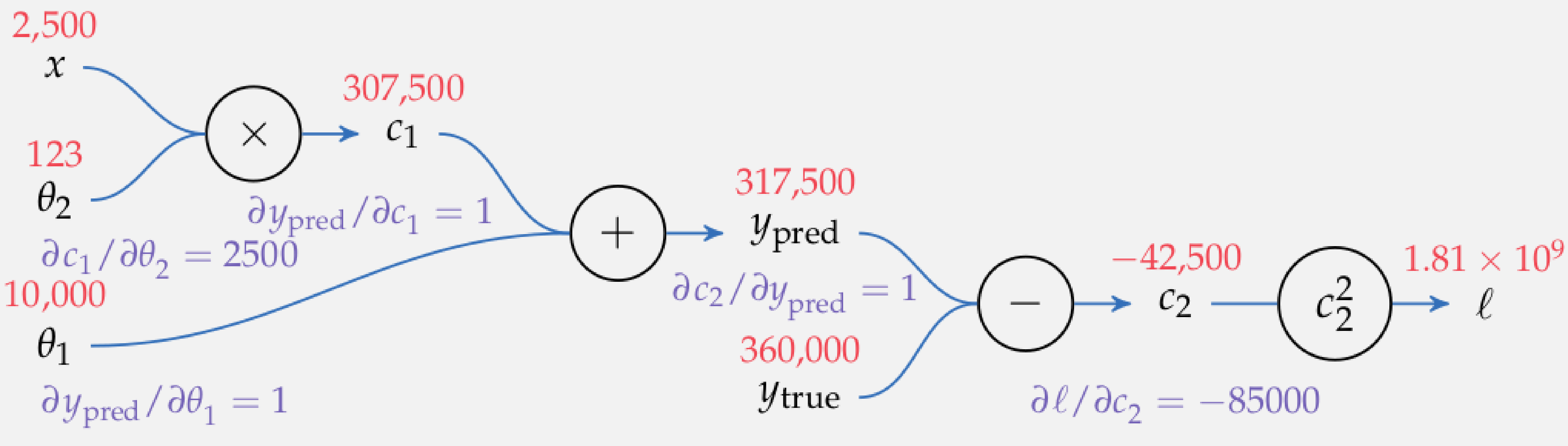


$$\theta^* = \arg\min_\theta \sum_{(x,y)\in\mathcal{D}} l(f_\theta(x), y)$$

Stochastic Gradient Descent: $\theta \leftarrow \theta - \alpha \nabla_\theta l(f_\theta(x), y)$

Chain Rule

Backprop



$$\frac{\partial \ell}{\partial \theta_1} = \frac{\partial \ell}{\partial c_2} \frac{\partial c_2}{\partial y_{\text{pred}}} \frac{\partial y_{\text{pred}}}{\partial \theta_1} = -85{,}000 \cdot 1 \cdot 1 = -85{,}000$$

$$\frac{\partial \ell}{\partial \theta_2} = \frac{\partial \ell}{\partial c_2} \frac{\partial c_2}{\partial y_{\text{pred}}} \frac{\partial y_{\text{pred}}}{\partial c_1} \frac{\partial c_1}{\partial \theta_2} = -85{,}000 \cdot 1 \cdot 1 \cdot 2500 = -2.125 \times 10^8$$

a “fast and furious” approach to training neural networks does not work and only leads to suffering. Now, suffering is a perfectly natural part of getting a neural network to work well, but it can be mitigated by being thorough, defensive, paranoid, and obsessed with visualizations of basically every possible thing. The qualities that in my experience correlate most strongly to success in deep learning are patience and attention to detail.

- Andrej Karpathy

Adaptive Step Size: RMSProp

Adaptive Step Size: ADAM

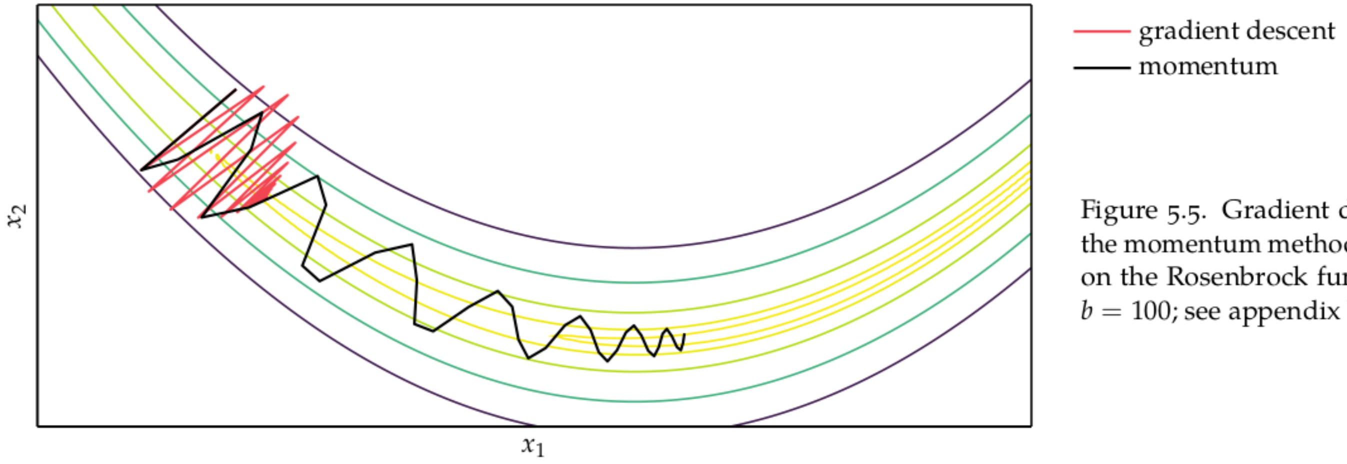
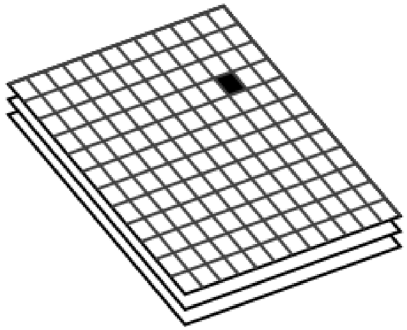
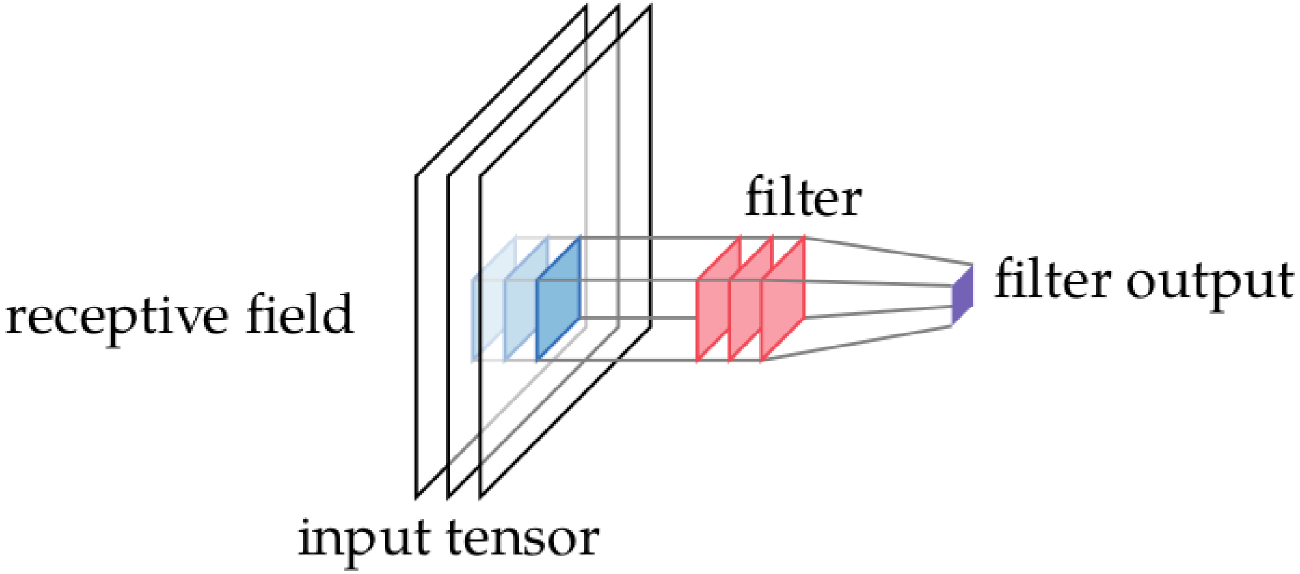
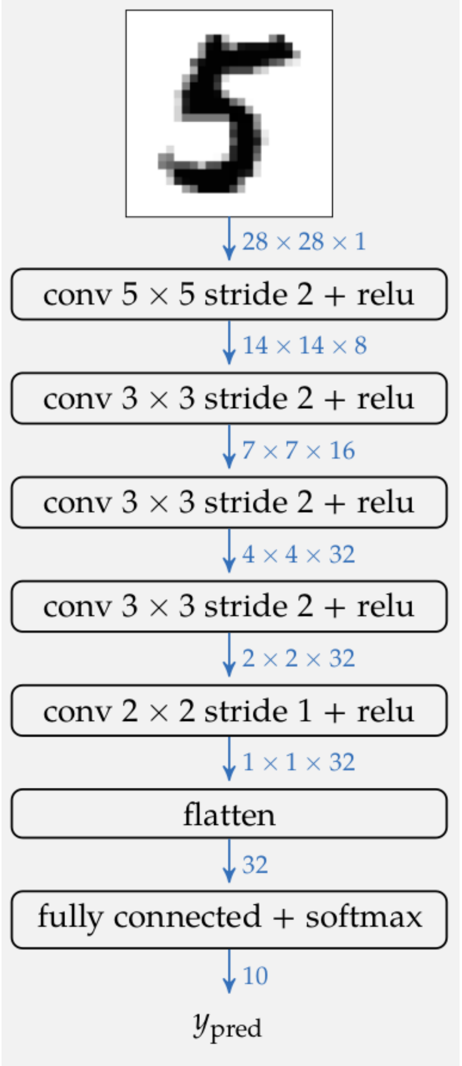


Figure 5.5. Gradient descent and the momentum method compared on the Rosenbrock function with $b = 100$; see appendix B.6.

On Your Radar: ConvNets







On Your Radar: Regularization

$$\arg\min_{\theta} \sum_{(x,y)\in\mathbf{D}} \ell(f_{\theta}(x), y) - \beta\|\theta\|^2$$

e.g. Batch norm, layer norm, dropout

On Your Radar: Skip Connections (Resnets)

Resources

OpenAI Spinning up