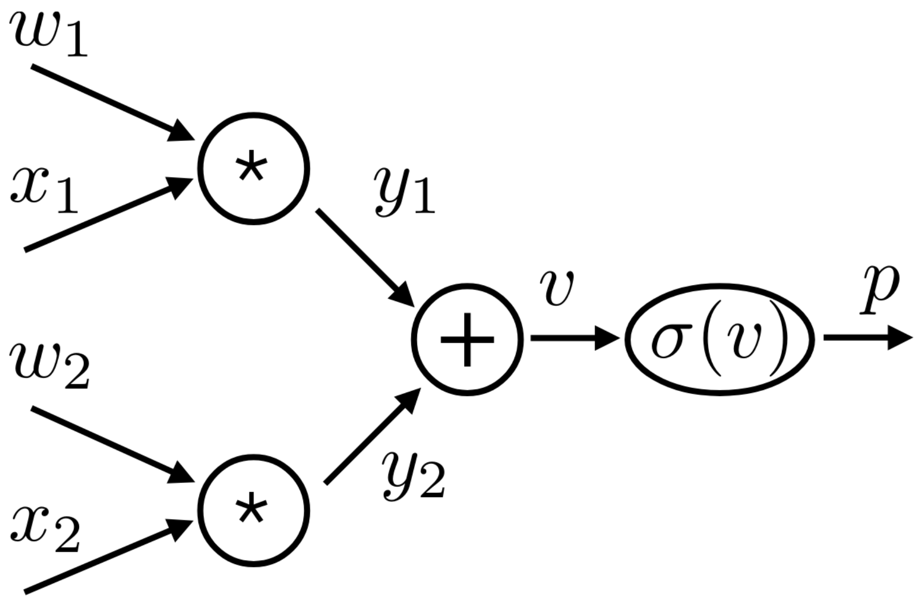
Autograd

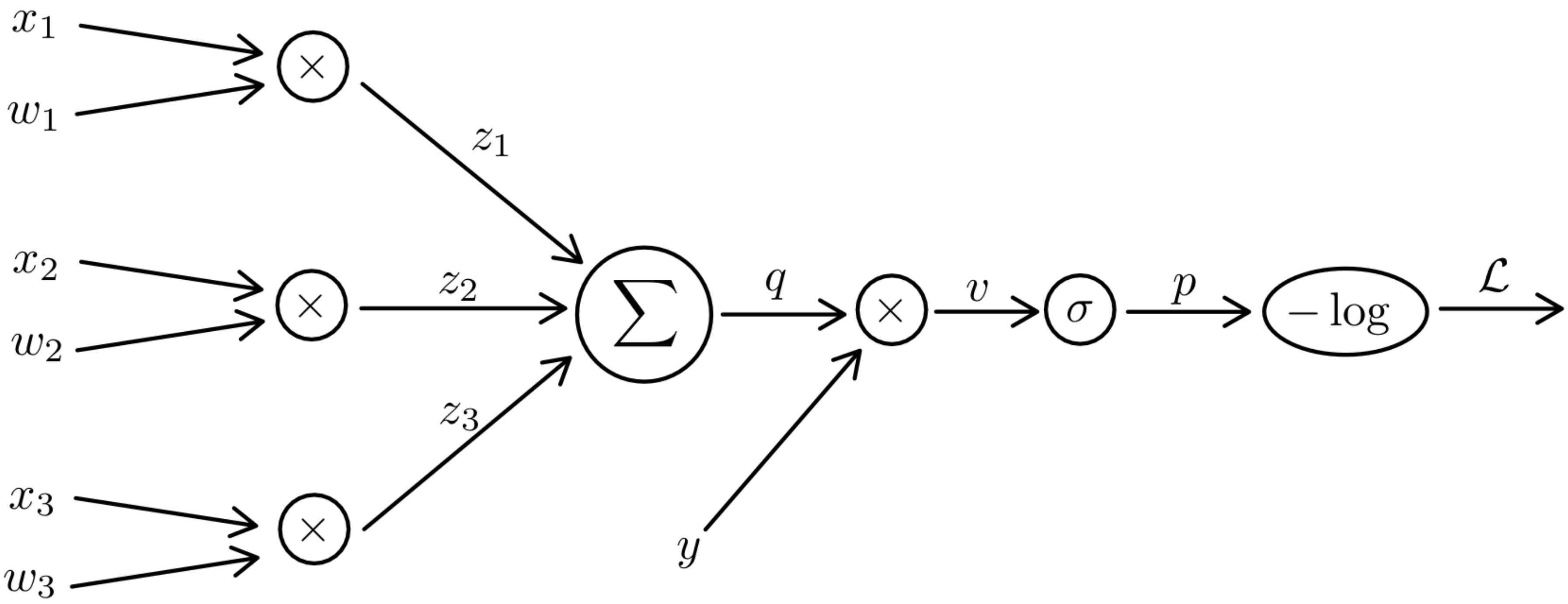
Computational graphs, backpropagation and the automatic gradient computation.

Jan Vlk
Czech Technical University in Prague
Faculty of Electrical Engineering, Department of Cybernetics

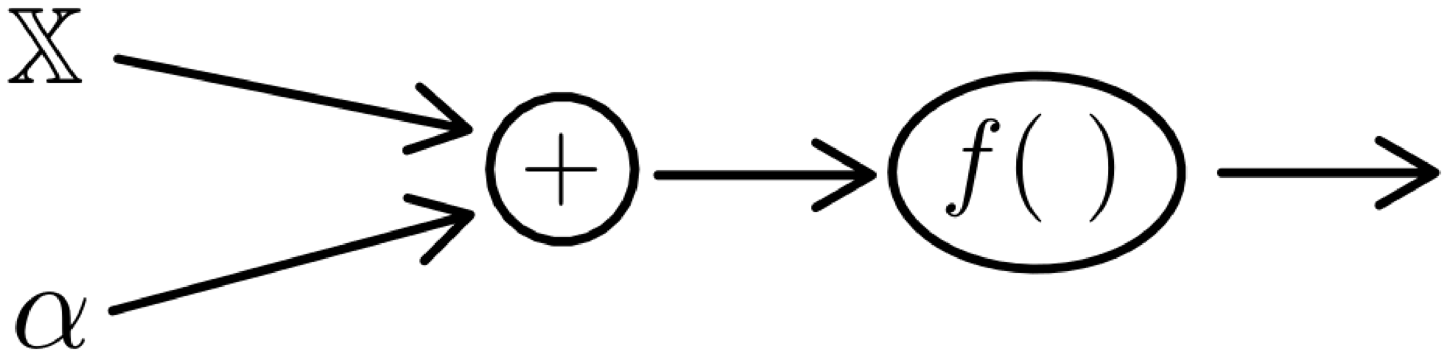
Computational graph



Training weights



Gradient in broadcasting



Backpropagation in code

Under the hood of autograd library

Activation functions

Sigmoid

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

$$\frac{\partial f}{\partial x} = \frac{e^{-x}}{(e^{-x}+1)^2}$$

ReLU

$$f(x) = max(0, x)$$

$$\frac{\partial f}{\partial x} = \begin{matrix} 1, x \geq 0 \\ 0, x < 0 \end{matrix}$$

Tanh

$$f(x) = \frac{1-e^{-2x}}{1+e^{-2x}}$$

$$\frac{\partial f}{\partial x} = \frac{4e^{2x}}{(e^{2x}+1)^2}$$

Logistic loss

$$f(x, y) = \log\left(1 + e^{-xy}\right)$$

$$\frac{\partial f(x, y)}{\partial x} = -\frac{y}{1 + e^{xy}}$$

HW1 Autograd

Creating your own library

Matrix multiplication

$$\begin{bmatrix} x_{11} & \cdots & x_{1n} \\ \vdots & \vdots & \vdots \\ x_{m1} & \cdots & x_{mn} \end{bmatrix} \times \begin{bmatrix} y_{11} & \cdots & y_{1m} \\ \vdots & \vdots & \vdots \\ y_{n1} & \cdots & y_{nm} \end{bmatrix} = \begin{bmatrix} x_{11} \cdot y_{11} + \cdots + x_{1n} \cdot y_{n1} & \cdots & x_{11} \cdot y_{1m} + \cdots + x_{1n} \cdot y_{nm} \\ \vdots & \vdots & \vdots \\ x_{m1} \cdot y_{11} + \cdots + x_{mn} \cdot y_{n1} & \cdots & x_{11} \cdot y_{1m} + \cdots + x_{1n} \cdot y_{nm} \end{bmatrix}$$

$$\frac{\partial f}{\partial Y} = X^T \qquad \frac{\partial f}{\partial X} = Y^T$$

Backward pass in code

$$vjp_f(v, X) = v \times \frac{\partial f}{\partial X} = v \times Y^T \qquad vjp_f(v, Y) = \frac{\partial f}{\partial Y} \times v = X^T \times v$$

https://math.stackexchange.com/questions/1846339/why-does-the-gradient-of-matrix-product-ab-w-r-t-a-equal-bt

Reqularization loss

$$f(x, y) = y \cdot \sum_i x_i^2$$

$$y \in R$$

$$\frac{\partial f}{\partial x_i} = 2 \cdot y \cdot x_i$$

Cross-entropy loss

$$f(x, y) = -\sum_{c \in Y} [\![y = c]\!] \log(x) \qquad \frac{\partial f}{\partial x} = -\sum_{c \in Y} [\![y = c]\!] \frac{1}{x}$$

x is a matrix with class probabilities for inputs
y is a vector of correct class predictions for inputs
Y is a matrix of all possible classes

Cross-entropy loss

$$f(x,y) = -\sum_i y_i \log(a_i), a_i = h(x_i) = \frac{e^{x_i}}{\sum_j e^{x_j}}$$

$$\frac{\partial f}{\partial x_i} = \sum_j \frac{\partial f}{\partial a_j} \cdot \frac{\partial a_j}{\partial x_i}$$

$$\frac{\partial f}{\partial x_i} = \sum_{j \neq i} \frac{\partial f}{\partial a_j} \cdot \frac{\partial a_j}{\partial x_i} + \frac{\partial f}{\partial a_i} \cdot \frac{\partial a_i}{\partial x_i}$$

$$\frac{\partial f}{\partial x_i} = \sum_{j \neq i} y_j \cdot a_i - y_i(1-a_i) = \sum_{j \neq i} y_j \cdot a_i + y_i \cdot a_i - y_i$$

$$\frac{\partial f}{\partial x_i} = \sum_j y_j \cdot a_i - y_i = a_i \sum_j y_j - y_i$$

$$\frac{\partial f}{\partial x_i} = a_i - y_i$$

$$\frac{\partial f}{\partial a_i} = \frac{\partial\left(-\sum_j y_j \cdot \log(a_j)\right)}{\partial a_i} = \frac{\partial - y_i \cdot \log(a_i)}{\partial a_i} = -\frac{y_i}{a_i}$$

$$\frac{\partial a_i}{\partial x_i} = \frac{\partial\left(\frac{e^{x_i}}{\sum_j e^{x_j}}\right)}{\partial x_i} = \frac{e^{x_i}}{\sum_j e^{x_j}}\left(\frac{e^{x_i}}{e^{x_i}} - \frac{e^{x_i}}{\sum_j e^{x_j}}\right) = a_i(1-a_i)$$

$$\frac{\partial f}{\partial a_j} = \frac{\partial\left(-\sum_k y_k \cdot \log(a_k)\right)}{\partial a_j} = -\frac{y_j}{a_j}$$

$$\frac{\partial a_j}{\partial x_i} = \frac{\partial\left(\frac{e^{x_j}}{\sum_k e^{x_k}}\right)}{\partial x_i} = \frac{e^{x_j}}{\sum_k e^{x_k}}\left(\frac{\frac{\partial e^{x_j}}{\partial x_i}}{e^{x_j}} - \frac{\frac{\partial \sum_k e^{x_k}}{\partial x_i}}{\sum_k e^{x_k}}\right) = \frac{e^{x_j}}{\sum_k e^{x_k}}\left(0 - \frac{e^{x_i}}{\sum_k e^{x_k}}\right) = -a_j \cdot a_i$$

https://towardsdatascience.com/deriving-backpropagation-with-cross-entropy-loss-d24811edeaf9

Direct kinematic task

Use of backpropagation