eCornell

CHEAT SHEET

Neural Network Cheat Sheet

Algorithm Name	Neural networks
Description	Neural networks are versatile models. They use piecewise linear functions to approximate decision boundaries (classification) or the label function (regression).
Applicability	Any supervised learning problem (classification or regression)
Assumptions	Input features are homogeneous. Label function is smooth.
Underlying Mathematical Principles	Transition function Loss function Forward propagation Backward propagation Stochastic gradient descent
Transition Functions	ReLU = $\sigma(z) = \max(z, 0)$ Sigmoid = $\sigma(z) = \frac{1}{1+e^{-z}}$ tanh = $\sigma(z) = \tanh(z)$
Loss Functions	**Regression:** Squared loss **Classification:** Cross entropy loss
Additional Details	• Training is usually done using SGD — a variant of GD that uses gradient computed from a small batch of training examples (sampled in each iteration). • Gradient is computed efficiently using backwards propagation. • Weight decay and dropout are used to regularize the model.

Pseudocode

Forward Propagation (x):

$\mathbf{z}_0 \leftarrow \mathbf{x}$;
for ℓ=*1:L* **do**
  $\mathbf{a}_\ell = \mathbf{w}_\ell^\top \mathbf{z}_{\ell-1} + b_\ell$;
  $\mathbf{z}_\ell = \sigma_e ll(a_\ell)$;
end
return $(\mathbf{z}_L)$

Backward Propagation
$(\{\mathbf{W}_1 \dots \mathbf{W}_L\}, \{b_1 \dots b_L\}, \{\mathbf{z}_1 \dots \mathbf{z}_L\}, \{\mathbf{a}_1 \dots \mathbf{a}_L\}, \mathcal{L})$
$\vec{\delta}_L \leftarrow \frac{\partial \mathcal{L}}{\partial \mathbf{z}_L} \odot \sigma'_\ell(\mathbf{a}_L)$;
for ℓ=*1:L* **do**
  $\mathbf{W}_\ell = \mathbf{W}_\ell - \alpha \vec{\delta}_\ell \mathbf{z}_{\ell-1}^T$;
  $b_\ell = b_\ell - \alpha \delta_\ell$;
  $\vec{\delta}_{\ell-1} = \sigma'_{\ell-1}(\mathbf{a}_{\ell-1}) \odot \left(\mathbf{W}_\ell^T \vec{\delta}_\ell\right)$;
end