Neural Networks in the Belle II Track Trigger

Ringberg Young Scientist Workshop 2016

June 8, 2016

Neuro team
S. Bähr, C. Kiesling, S. Neuhaus, S. Skambraks
estimate z-vertex of single tracks to reject machine background in Belle II

Why neural networks?
- fast \((< 1 \mu s)\), parallel, deterministic runtime
- nonlinear, noise robust
- ...
The Neurotrigger

estimate z-vertex of single tracks to reject machine background in Belle II

Why neural networks?

- fast (< 1 μs), parallel, deterministic runtime
- nonlinear, noise robust
- ...

How does it work?

- multi layer perceptron
- function approximation
- classification
- training methods
- application in the neurotrigger
Neural Networks in the Belle II Track Trigger
Multi Layer Perceptron (MLP)

- **Input layer**
- **≥ 1 hidden layers**
- **Output layer**
Multi Layer Perceptron (MLP)

Neural Networks in the Belle II Track Trigger
General function approximation

**Fourier series:**
\[
f(\vec{x}) = \sum_k c_k \cdot \exp(2\pi i \vec{k} \cdot \vec{x})
\]

**Hidden node:**
\[
tanh(\sum_i w_{1ij} x_i) = \tanh(\beta_j (u(\vec{x}) - \alpha_j))
\]

**Output node:**
\[
\sum_j w_{2jk} \cdot \tanh(\beta_j (u(\vec{x}) - \alpha_j)) \approx \sin(u(\vec{x}))
\]
Classification: XOR

need at least 2 separating lines = hidden nodes

a XOR b

Neural Networks in the Belle II Track Trigger
Training with backpropagation

training samples \((x^s, t^s)\)

\[ \rightarrow \text{minimize cost function } E = \sum_{k,s} (z_k(x^s) - t^s_k)^2 = \sum_{k,s} (\Delta^s_k)^2 \]

weight updates: \( \Delta w = -\frac{\partial E}{\partial w} \cdot \delta, (\delta: \text{learning rate}) \)
Training with backpropagation

training samples \((\vec{x}^s, \vec{t}^s)\)

\[ \rightarrow \text{minimize cost function } E = \sum_{k,s} (z_k(\vec{x}^s) - t_k^s)^2 = \sum_{k,s} (\Delta_k^s)^2 \]

\text{weight updates: } \Delta w = -\frac{\partial E}{\partial w} \cdot \delta, \ (\delta: \text{learning rate})
Training with backpropagation

training samples \((\vec{x}^s, \vec{t}^s)\)

\[ \rightarrow \text{minimize cost function } E = \sum_{k,s} (z_k(\vec{x}^s) - t^s_k)^2 = \sum_k (\Delta^s_k)^2 \]

weight updates: \[ \Delta w = -\frac{\partial E}{\partial w} \cdot \delta, \quad (\delta: \text{learning rate}) \]

\[
\begin{align*}
\frac{\partial E}{\partial w_{1ij}} &= y'_j x_i \\
\sum_i w_{1ij} x_i &= z_k y_j \\
\sum_j w_{2jk} y_j &= \Delta_k \\
y'_j &= \tanh'(\sum_i \ldots) \cdot \sum_k w_{2jk} z'_k 
\end{align*}
\]
RPROP: flexible learning rate

**weight updates:** $\Delta w = -\text{sign} \left( \frac{\partial E}{\partial w} \right) \cdot \delta_{\text{epoch}}(w)$

- depends only on sign of cost function derivative
- individual learning rate for each weight
- learning rate adjusted during training

\[
\left( \frac{\partial E}{\partial w} \right)^{\text{epoch}} \cdot \left( \frac{\partial E}{\partial w} \right)^{\text{epoch}-1} > 0 : \quad \left( \frac{\partial E}{\partial w} \right)^{\text{epoch}} \cdot \left( \frac{\partial E}{\partial w} \right)^{\text{epoch}-1} < 0 :
\]

$\rightarrow$ increase $\delta(w)$

$\rightarrow$ decrease $\delta(w)$
Overtraining

- Training epoch
- Cost function

Training set vs. Validation set

- Independent validation set to avoid “learning by heart”
Overtraining

- Training epoch
- Cost function
- Training set
- Validation set

Independent validation set to avoid “learning by heart”
Task: 2D track (circle) → 3D track (helix)

- Input: wire hits from the central drift chamber
  - 9 super layers, alternating axial and stereo
  - 1 hit per super layer, scaled position and drift time
- Output: continuous scaled z-vertex

Axial layer

Stereo layer
What does the MLP learn?

SL1 turned off

SL3 turned off

SL5 turned off

SL7 turned off

zNN - zMC [cm]

trained without noise

trained with noise

Neural Networks in the Belle II Track Trigger
Multi Layer Perceptron

- $\geq 3$ layers of perceptrons
- connection weights, nonlinear activation
- function approximation / classification

Training: backpropagation

- input/output samples
- minimize cost function

Belle II neurotrigger

- fast 3D track reconstruction