Conventional ANNs

- Inspired by the human brain
- Benchmarks on tasks solved by humans
- ...but compute in a fundamentally different way compared to the biological brain
- Lack a **time dimension**

![Conventional ANN diagram](image1)

---

Biological networks

- Neuron spikes are **discrete events**
- Asynchronous
- Can encode information in temporal patterns of activity
- Energy-efficient

![Neuron spike waveform](image2)

Source: https://en.wikipedia.org/wiki/Electrophysiology
Information coding in biological brains

- Conventional approach: rate coding
  - *slowly* accumulate over spikes

- Alternative: temporal coding
  - single spikes at **precise times**
  - fast but possibly less accurate

- Information is carried by relative spike times
  - retinal ganglion cells encode the spatial structure of an image in the relative timing of their first spikes (Gollish & Meister, 2008)
  - tactile afferents encode information about fingertip events in the relative timing of the first spikes (Johansson & Birznieks, 2004)

- Single spikes carry information across brain areas
  - 10 synaptic stages crossed within 100ms in the visual system, suggesting that responses are made on the basis of single spikes (Thorpe & Imbert, 2016)

*Information can be encoded in the timing of individual spikes.*
Artificial networks with temporal coding

- Earlier spikes encode more salient information
- Consider a classification problem with $m$ inputs and $n$ possible classes
- Temporal encoding
  - $m$ input neurons
  - spike at time proportional to brightness of corresponding input pixel
- Temporal decoding
  - $n$ output neurons
  - class $k$ iff $k$th neuron spikes earliest
**Building a spiking model**

- Alpha synaptic function: $t \cdot e^{-\tau t}$
- Custom decay: $\tau$
- Weighted presynaptic inputs: $w_i$
- Fixed spiking threshold: $\theta$
- May not spike
- Allows forgetting inputs
- Richer dynamics than nonleaky models
- Computing the spike time given a set $I$ of inputs:

\[
V(t) = \sum_{i \in I} w_i(t - t_i)e^{\tau(t_i - t)}
\]

\[
t_{\text{out}} = \frac{B_I}{A_I} - \frac{1}{\tau}W\left(-\frac{\theta}{A_I} e^{\tau t} \sum_{i \in I} w_i e^{\tau t_i} \right)
\]

where

\[
A_I = \sum_{i \in I} w_i e^{\tau t_i}
\]

\[
B_I = \sum_{i \in I} w_i e^{\tau t_i t}
\]
Learning with backpropagation

- Usual problem with spiking networks: non-differentiable spike events
- Learning goal with temporal coding: adjust the timing of outputs
- Postsynaptic spike times depend on presynaptic spike times and their weights

\[
\frac{\partial t_{out}}{\partial t_j} = \frac{w_j e^{\tau t_j}((t_j - \frac{B_l}{A_l})\tau + W_I + 1)}{A_l(1 + W_I)} \quad \frac{\partial t_{out}}{\partial w_j} = \frac{e^{\tau t_j}(t_j - \frac{B_l}{A_l} + \frac{W_I}{\tau})}{A_l(1 + W_I)} \quad \text{where} \quad W_I = W(-\frac{\theta}{A_l} e^{x_I})
\]

- Minimize cross-entropy loss with Adam optimizer
- At the output layer, minimize spike time: softmax of negative spike times
Synchronization pulses

- A set of neurons connected to each non-input layer
- Act like temporal biases
- Ensure there are spikes (eventually)
- Learnable spike times and weights
## MNIST experiment

<table>
<thead>
<tr>
<th>Hyperparameter</th>
<th>Search space</th>
<th>Chosen value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decay constant $\tau$</td>
<td>$[0.1, 2]$</td>
<td>0.18</td>
</tr>
<tr>
<td>Fire threshold $\theta$</td>
<td>$[0.1, 1.5]$</td>
<td>1.17</td>
</tr>
<tr>
<td>Number of hidden layers.</td>
<td>$[0, 4] \times [2, 1000]^{*}$</td>
<td>$1 \times 340$</td>
</tr>
<tr>
<td>Number of pulses per layer.</td>
<td>$[0, 10]$</td>
<td>10</td>
</tr>
<tr>
<td>Multiplier for non-pulse weights initialization.</td>
<td>$[-10, 10]$</td>
<td>-0.275</td>
</tr>
<tr>
<td>Multiplier for pulse weights initialization.</td>
<td>$[-10, 10]$</td>
<td>7.84</td>
</tr>
<tr>
<td>Learning rate for network weights.</td>
<td>$[10^{-5}, 1]^{*}$</td>
<td>$10^{-4} \times 2.02$</td>
</tr>
<tr>
<td>Learning rate for pulse timings.</td>
<td>$[10^{-5}, 1]^{*}$</td>
<td>$10^{-2} \times 5.95$</td>
</tr>
<tr>
<td>Mini-batch size for Adam optimization.</td>
<td>$[1, 1000]^{*}$</td>
<td>5</td>
</tr>
<tr>
<td>Clipping value for derivatives.</td>
<td>$[1, 1000]$</td>
<td>539.7</td>
</tr>
<tr>
<td>Penalty added to presynaptic weights if a neuron didn’t fire.</td>
<td>$[0, 100]$</td>
<td>48.38</td>
</tr>
</tbody>
</table>

* - logarithmic search space
MNIST experiment

- Network size $784 \times 340 \times 10$ (plus 10 synchronization pulses)
- MNIST digits were encoded as spikes at times between 0 and 1
- Pulses were initialized to spike at evenly distributed times between 0 and 1

- Results: 97.96% accuracy on MNIST test
- For comparison: a non-convolutional ReLU DNN achieves 97.9%
Slow and fast classification regimes

During training, the same network *spontaneously* switched between two operating regimes:

- slow but more accurate
- very fast but less accurate

<table>
<thead>
<tr>
<th></th>
<th>Slow regime</th>
<th>Fast regime</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training accuracy (%)</td>
<td>99.9633</td>
<td>99.885</td>
</tr>
<tr>
<td>Training loss (mean)</td>
<td>0.002884</td>
<td>0.00444</td>
</tr>
<tr>
<td>Test accuracy (%)</td>
<td>97.96</td>
<td>97.4</td>
</tr>
<tr>
<td>Test loss (mean)</td>
<td>0.173248</td>
<td>0.19768</td>
</tr>
</tbody>
</table>

The same speed-accuracy trade-off is observed in human decision making.
Take-away points

● The timing of single spikes efficiently encodes information in biological brains.

● Spiking networks with temporal coding:
  ○ can be trained with backpropagation
  ○ can perform digit recognition at competitive accuracies

● Interesting from a multidisciplinary perspective:
  ○ shed light on the representational capabilities of biological-like networks
  ○ possible model for efficient neuromorphic computing

● Opening pathways towards spiking nets research:
  ○ recurrent, state-based spiking networks that perform efficient computation
Open-source code available
https://github.com/google/ihmehimmeli

Thank you!

Iulia Comșa    Krzysztof Potempa    Luca Versari    Thomas Fischbacher    Andrea Gesmundo    Jyrki Alakuijala