Opening the Black Box – Data Driven Visualizaion of Neural Networks

Fan-Yin Tzeng & Kwan-Liu Ma

September 20, 2006

Artificial Neural Networks

- Artificial Neural Networks (ANNs) mimic the processes found in biological neural networks.
  - The brain is a massively parallel information processing system formed by about ten billion nerve cells (neurons) and their synapses.
- Used for predicting and learning from a given set of data.
- Based on the combination of neurons, connections, transfer functions, learning algorithms, and neuron layout methods.

Limitations of ANNs

- ANNs act as a black box.
  - Information stored is a set of weights and connections that provides no insight as to how a task is performed.
  - The relationship between the input and output is not clear.
- ANNs do not provide a solution that is easily understood or validated.
- Parameter selection is tedious, and often done in a trial-and-error process.

Use of Visualization

- Visualizations of ANNs give insight to how the NN works.
- Viz also used in network structuring and parameter selection, saving time and cost.
- Previous ANN viz did not focus on data used by network.
  - This paper takes a data driven approach.
  - In addition to ANN viz, the user is allowed to interact with data and see the effect on the network.

In General

- Each node (neuron) is connected to all other nodes in the adjacent layer.
- Each connection between nodes has a weight, with the weights modulating the value across the connection.
- Notation:
  - Nodes in input layer: $I_1, I_2, I_3, \ldots, I_m$
  - Nodes in Hidden layer: $H_1, H_2, H_3, \ldots, H_n$
  - Weight of connection between I_i and H_j: W_{ij}
- The value of a node in the hidden layer is given by: $H_j = TF(\sum_{i=1}^{n} W_{ij} \times I_i)$

(Cont.)

- The value of a node in the output layer (O_k):
  $O_k = TF(\sum_{i=1}^{m} W_{jk} \times H_j)$
- $TF(x)$ is a non-linear transfer function.
  - Transfer function is what converts NN to a non-linear system.
  - Non-linear needed since classifying NNs make terminal desicions.
  - When calculating the value of an output node, the same transfer function used on the previous layer is applied to the summed results from that (previous) layer.

Training

- At the beginning....
  - We need a set of training inputs and desired outputs.
  - Weights are set at random.
- Weights are then iteratively modified to obtain minimum error.
- The criterion for "minimum" is some arbitrary threshold (maybe a small one).
- After training, the network may be used for data similar to the training set.

Specifics

- NN is feed-forward.
  - Input always moves from left to right, no loops.
- Back-propogation training algorithm used.
  - Mean squared error calculated first calculated at output level for a given input.
  - Error is then minimized, by changing weights, in a back-to-front manner at the next level.
  - Error minimization iteratively continues until threshold reached.
  - Repeat steps 1-3 on the other test data.
  - Start all over again.

(Cont.)

- The standard sigmoid function is used for the transfer function.
  - $f(x) = \frac{1}{(1+e^{-x})}$
  - A continuous differentiable version of a stair-stepping function.
  - Is the most common transfer fucntion used for classification.

Volume Classification

- Trained by a small set of input data including the corresponding class IDs.
- Input vector includes a voxel's scalar value, gradient magnitude, its neighbors' scalar values, and its position.

Spam Classification

- Trained by a set of pre-classified spam and non-spam email messages (117, 283 respectively).
- Words and phrases from emails form input vectors.
- This NN used because of its large size (82 input nodes and 100 hidden nodes).

Single Data

- Probe is provided to select from data domain.
- NN is then shown as data are passing through the network.
- Input and output nodes are colored based on the selected voxel's value.
- A connection's width is based on its importance.
  - The focus is on the input-hidden layer, so results can be mapped to input data domain.
  - Importance based on weight.
  - The layer's influence is propogated by multiplying each weight between both layers that connect to the same hidden node.

Set of Data

- A region of data is now selected.
- Now, input node size is also set, also based on importance, or contribution of the node to the result.
- For large NNs, the nodes can be arranged by order of importance.
- Relative importance is calculated by a ratio of weights associated with that node to that of all nodes (in the same layer).
- The user can (maybe) set a threshold that will hide less important connections.

(Cont.)

- The assumption is made that selected data are similar (values are close to the mean of values), so weights between input-hidden layer are divided by those means.
- The standard deviation is also calculated, and nodes are assigned color based on the similarity of mean and standard deviation.

Visualizing Uncertainty

- NN outputs a value representing uncertainty.
  - High and low values indicate low uncertainty.
  - Middle values indicate hight uncertainty.
- Parallel coordinates are used to show the inputs and outputs when training or classifying.
  - Colors correspond to how well the NN classified certain sets of data.
  - This can be used to understand what type of test data is needed.
- Boundaries between materials, due to interpolation will appear (by color) to have a high level of uncertainty.
- The fix for this is to assign a voxel's color based on the value of the neighboring voxel along the opposite normal direction.

- None listed!
- The authors do note that based on their rendered node/connection weights for a particular NN, 4 nodes were removed from the hidden layer.
  - Cost of classification was reduced by 15%
  - Result 0.5% different from original result.