Biological Neuron

- Dendrite
- Cell body or Soma
- Nucleus
- Synapse
- Axon
- Axon from another cell
- Axonal arborization
- Synapses
Organization of Levels in Brains

- map into cerebral cortex, pathways, columns, topographic maps; involve multiple regions
- neurons of similar and different properties, 1 mm in size, localized region in the brain
- 100μm in size, contains several dendrite trees
Biological Analogy

- Brain Neuron
- Artificial neuron (processing element)
- Set of processing elements (PEs) and connections (weights) with adjustable strengths
ANN: History

- Pavlov’s conditioning experiments: a conditioned response, salivation in response to the auditory stimulus
- Lots of activities concerning automatas, communication, computation, understanding of nervous system during 1930s and 1940s
- McCulloch and Pitts 1943
- von Neumann EDVAC (Electronic Discrete Variable Automatic Computer)
- Hebb: The Organization of Behavior, 1949
- Minsky: Theory of Neural-Analog Reinforcement Systems and Its Application to the Brain-Model Problem (Reinforcement learning), 1954
- Uttley: leaky integrate and fire neuron, 1956
- Rosenblatt: the perceptron, 1958
ANN: History

- Long-Term Potential, LPT, (1973 Bliss, Lomo), AMPA receptor, Long-Term Depression, LTD, NMDA receptor,
- The nearest neighbor rule by Fix and Hodges 1951
- Least mean square algorithm by Widrow and Hoff in 1960
- The use of stochastic gradient in adaptive pattern classification by Amari in 1967
- The idea of competitive learning: von der Malsburg 1973, the self-organization of orientation-sensitive nerve cells in the striate cortex
- Self-organized maps by Grossberg in 70s
- Saving units (associative networks) by Anderson and Kohonen in 1982
- Recurrent neural networks by Hopfield in 1982
- Error backpropagation learning algorithm by Rumelhart, Hinton and Williams in 1986
- Spike timing dependence plasticity by Markram in 1997
- Echo state networks by Jaeger in 2002
- Lots of applications
- ...
(Artificial) Neural Network?

- Computational model inspired from neurological model of brain
- Human brain computes in different way from digital computer
  - highly complex, nonlinear, and parallel computing
  - many times faster than a computer in
    - pattern recognition, perception, motor control
  - has great structure and ability to build up its own rules by experience
    - dramatic development within 2 years after birth
    - continues to develop afterward
      - Language Learning Device before 13 years old
  - Plasticity: ability to adapt to its environment
Neural Network Definitions

• Machine designed to model the way in which brain performs tasks
  ▫ implemented by electronic devices and/or software (simulation)
  ▫ Learning is the major emphasis of NN
• Massively parallel distributed processor
  ▫ massive interconnection of simple processing units
  ▫ simple processing units store experience and make it available to use
  ▫ knowledge is acquired from environment thru learning process
• Learning Machine
  ▫ modify synaptic weights to obtain design objective
  ▫ modify own topology - neurons die and new one can grow
• Connectionist network - connectionism
Benefits of Neural Networks (I)

- Power comes from massively parallel distributed structure and learn to generalize
  - generalization: ability to produce reasonable output for inputs not encountered during training
- NN cannot provide solution by working individually
  - Complex problem is decomposed into simple tasks, and each task is assigned to a NN
  - Long way to go to build a computer that mimics human brain
- Non-linearity
  - interconnection of non-linear neurons is itself non-linear
  - desirable property if underlying physical mechanism is non-linear
Benefits of Neural Networks (II)

- Input-Output Mapping
  - input-output mapping is built by learning from examples
    - reduce differences of desired response and actual response
  - non-parametric statistical inference
    - estimate arbitrary decision boundaries in input signal space

- Adaptivity
  - adapt synaptic weight to changes of environment
  - NN is retrained to deal with minor change in the operating environment
    - change synaptic weights in real-time
  - more robust, reliable behavior in non-stationary environment
  - Adaptive pattern recognition, Adaptive signal processing, Adaptive control
  - stability-plasticity dilemma
Benefits of Neural Networks (III)

- **Evidential Response**
  - not only selected class label but also confidence
  - confidences can be used to reject
    - recognition accuracy vs. reliability (do only what you can do)

- **Contextual Information processing**
  - (contextual) knowledge is presented in the structure
  - every neuron is affected by others

- **Fault Tolerance**
  - performance degrades gracefully under adverse condition
    - catastrophic failure of digital computer

- **VLSI implementability**
  - massively parallel nature makes it well suited for VLSI implementation
Benefits of Neural Networks (IV)

- Uniformity of Analysis and Design
  - Neuron is common to all NN
  - share theories and learning algorithms
  - modular networks can be built thru seamless integration
- Neurobiological Analogy
  - living proof of fault tolerant, fast, powerful processing
  - Neuroscientists see it as a research tool for neurobiological phenomena
  - Engineers look to neuroscience for new ideas
**ANN: Architectures**

### Perceptron

- **Inputs**: 5, 3, 2, 5, 3, 5, 3, 2, 5, 3, 1, 0, 0, 1, 0, 5, 3, 2, 2, 1
- **Epoch**: Exemplar

### Multiple Layer Feedforward

- **Inputs**: 5, 3, 2, 5, 3, 5, 3, 2, 5, 3, 1, 0, 0, 1, 0, 5, 3, 2, 2, 1
- **Hidden Layer**: PEs
- **Output Layer**: PEs
- **Weights**: 5, 3, 2, 5, 3, 5, 3, 2, 2, 1

### Recurrent/Feedback

- **Inputs**: 5, 3, 2, 5, 3, 5, 3, 2, 5, 3, 5, 3, 2, 2, 1

### Time Lag Feedforward

- **Inputs**: 5, 3, 2, 5, 3
- **Memory Structure**: Mem
ANN: What Makes them “Unique”

• Neural networks are nonlinear models
  ▫ Many other nonlinear models exist
    • mathematics required is usually involved or nonexistent.
  ▫ simplified nonlinear system
  ▫ combinations of simple nonlinear functions

• Neural networks are trained from the data
  ▫ No expert knowledge is required beforehand
  ▫ They can learn and adapt to changing conditions online

• They are universal approximators
  ▫ learn any model given enough data and processing elements

• They have very few formal assumptions about the data
  ▫ (e.g. no Gaussian requirements, etc.)
**ANN: How do neural nets work?**

**TRAIN THE NETWORK:**
1. Introduce data
2. Computes an output
3. Output compared to desired output
4. Weights are modified to reduce error

**USE THE NETWORK:**
1. Introduce new data to the network
2. Network computes an output based on its training
ANN: Generalization

- Neural networks are very powerful, often **too** powerful.
- Can overtrain a neural network:
  - will perform very well on data that it was trained with
  - but poorly on test data
- Never judge a network based upon training data results ONLY!
The most common solution to the “generalization” problem is to divide your data into 3 sets:

- **Training data:**
  - used to train network

- **Cross Validation data:**
  - used to actively test the network during training - used to stop training

- **Testing data:**
  - used to test the network after training

- **Production data:**
  - desired output is not known (implementation)
Models of Neuron

- Neuron is information processing unit

- A set of synapses or connecting links
  - characterized by weight or strength

- An adder
  - summing the input signals weighted by synapses
  - a linear combiner

- An activation function
  - also called squashing function
    - squash (limits) the output to some finite values
Nonlinear model of a neuron (I)

\[ v_k = \sum_{j=1}^{m} w_{kj} x_j + b_k \]

\[ y_k = \varphi(v_k) \]
Nonlinear model of a neuron (II)

\[ X_0 = +1 \]

\[ W_{k0} = b_k \text{ (bias)} \]

\[ x_1 \]
\[ W_{k1} \]

\[ x_2 \]
\[ W_{k2} \]

\[ \vdots \]
\[ \vdots \]

\[ x_m \]
\[ W_{km} \]

\[ \sum \]
\[ v_k \]

Activation function

\[ \varphi(.) \]

Output

\[ y_k \]

- Input signal
- Synaptic weights
- Summing junction

\[ v_k = \sum_{j=0}^{m} w_{kj} x_j \]

\[ y_k = \varphi(v_k) \]
Types of Activation Function

Threshold Function

Piecewise-linear Function

Sigmoid Function (differentiable)

\[ \varphi(v) = \frac{1}{1 + \exp(-av)} \]

\( a \) is slope parameter
Activation Function value range

Signum Function

Hyperbolic tangent Function

\( \varphi(v) = \tanh(v) \)
The McCulloch-Pitts Model

- McCulloch and Pitts (1943) produced the first neural network, which was based on their artificial neuron.
- The activation of a neuron is binary.
- The neuron either fires (activation of one) or does not fire (activation of zero).
- Neurons in a McCulloch-Pitts network are connected by directed and weighted paths.
The McCulloch-Pitts Model

• For the network shown below the activation function for unit Y is:
  \[ f(y_{in}) = 1, \text{ if } y_{in} \geq T \]
  \[ \text{else } 0 \]
  where \( y_{in} \) is the total input signal received and \( T \) is the threshold for Y.
Example: Logical Functions

- McCulloch and Pitts: some Boolean functions can be implemented with an artificial neuron (not XOR).
two-layer network capable of calculating XOR

\[ z = \text{XOR}(x, y) \]
Stochastic Model of a Neuron

- Deterministic vs stochastic
- stochastic: stay at a state with probability $P$

\[ x = \begin{cases} 
+1 & \text{with probability } P(v) \\
-1 & \text{with probability } 1 - P(v) 
\end{cases} \]

$x$: state of neuron  
$v$: induced local field (input sum)  
$P(v)$: probability of firing

\[ P(v) = \frac{1}{1 + \exp(-\frac{v}{T})} \]

where $T$ is pseudotemperature
$T \to 0$, reduced to deterministic form
NNs as directed Graphs

- Block diagram can be simplified by the idea of signal flow graph
- Node is associated with signal
- Directed link is associated with transfer function
  - **Synaptic links**
    - Governed by linear input-output relation
    - Signal $x_j$ is multiplied by synaptic weight $w_{kj}$
  - **Activation links**
    - Governed by nonlinear input-output relation
    - Nonlinear activation function
Signal Flow Graph of a Neuron

\[ x_0 = +1 \]

\[ W_{k0} = b_k \]

\[ x_1, w_{k1} \]

\[ x_2, w_{k2} \]

\[ \vdots \]

\[ x_m, w_{km} \]

\[ v_k \]

\[ \varphi(.) \]

\[ y_k \]
Architectural graph of a Neuron

- Partially complete directed graph describing layout

- Three graphical representations
  - Block diagram - providing functional description of a NN
  - Signal flow graph - complete description of signal flow
  - Architectural graph - network layout
Network Architecture

• Single-layer Feedforward Networks
  ▫ input layer and output layer
    • single (computation) layer
  ▫ feedforward, acyclic

• Multilayer Feedforward Networks
  ▫ hidden layers - hidden neurons and hidden units
  ▫ enables to extract high order statistics
  ▫ 10-4-2 network, 100-30-10-3 network
  ▫ fully connected layered network

• Recurrent Networks
  ▫ at least one feedback loop
  ▫ with or without hidden neuron
Network Architecture

Single layer

Multiple layer fully connected

Recurrent network with hidden units

Recurrent network without hidden units

Unit delay operator
Feedback

- Output is fed-back to the NN that is used in determining the output itself

\[ y_k(n) = \sum_{i=0}^{\infty} w^{i+1} x_j(n-l) \]

- depending on \( w \)
  - stable, linear divergence, exponential divergence
  - we are interested in the case of \(|w| < 1\); infinite memory
    - output depends on inputs of infinite past
- NN with feedback loop: recurrent network
Knowledge Representation

- Knowledge refers to *stored information or models* used by a person or machine to interpret, predict and appropriately respond to the outside world
  - What information is actually made explicit;
  - How the information is physically encoded for the subsequent use
- Good solution depends on good representation of knowledge
- In NN, knowledge is represented by internal network parameters
  - real challenge
- Knowledge of the world
  - world state represented by known facts - prior knowledge
  - observations - obtained by (noisy) sensors; training examples
Knowledge Acquisition by NN Training

- Training examples: either labeled or unlabeled
  - labeled: input signal and desired response
  - unlabeled: different realizations of input signal
  ▪ Examples represent the knowledge of environment
- Character recognition
  1. Appropriate architecture is selected for NN
     - source node = number of pixels of input image
     - e.g. 26 output node for each digit
     - subset of examples for training NN by suitable learning algorithm
  2. Recognition performance is tested by the rest of the examples
- Positive and negative examples
Classification: Optical Character Recognition

- Determine if the input image is the A, B, C, ...
  - 2 classes: create one output for each class (e.g., class 0: true or false, etc.).
  - 26 outputs (A...Z). Each image is labeled with a class
    - image A will be (1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
    - image B will be (0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0), etc.
- Must train the network to recognize the alphabets
Rules of Knowledge representation in NN

• Similar input from similar classes produce similar representations
  ▫ similarity measures
    • Euclidian distance, dot (inner) product, cos
    • random variable : Mahalanobis distance
    • ...

• Separate classes produce widely different representations

• More neurons should be involved in representation of more important feature
  ▫ probability of detection / false alarm

• Prior information and invariances should be built into the design of the network
  ▫ general purpose vs specialized
Building Prior to NN design

• Specialized structure
  ▫ learns fast because of small free parameters
  ▫ runs fast because of simple structure
• No well-defined rules for building specialized NN
  ▫ ad hoc approach
  ▫ restricting the network architecture through using local connections
    ▬ receptive field
  ▫ Constraining the choice of synaptic weights
    ▬ weight sharing, parameter tying
Building invariance to NN design

- Want to be capable to cope with transformations
  - **Invariance by structure**
    - synaptic connections are arranged not to be affected by transformation
    - rotation invariant forcing $w_{ji} = w_{jk}$ for all $k$ in the same distance from the center of image
  - **Invariance by training**
    - train by data of many different transformations
    - computationally infeasible
  - **invariant feature space**
    - use features invariant to the transformations
- No well-developed theory of optimizing architecture of NN
- NN lacks explanation capability
AI and NN

- Definition of AI; Goal of AI
  - art of creating machine that performs tasks that requires intelligence when performed by people
  - study of mental faculties through the use of computational models
  - to make computers to perceive, reason and act
  - to develop machine that perform cognitive tasks

- functions of AI system
  - store knowledge
  - apply the knowledge to solve problems
  - acquire new knowledge thru experience

- Key components of AI
  - representation
  - reasoning
  - learning
AI

- AI is goal, objective, dream
- NN is a model of intelligent system
  - it is not the only system
  - Intelligent system is not necessarily same as human
    - Example: Chess machine
- Symbolic AI is a tool, paradigm toward AI
- NN can be a good tool toward AI
Reasoning

• Reasoning is ability to solve problem
  ▫ must able to express and solve broad range of problems
  ▫ must able to make explicit and implicit information known to it
  ▫ must have control mechanism to select operators for a situation
• Problem solving is a searching problem
• deal with incompleteness, inexactness, uncertainty
  ▫ probabilistic reasoning, plausible reasoning, fuzzy reasoning
Learning

- Model of Machine Learning
  - Environment,
  - Learning element,
  - Knowledge base, and
  - performance cycle
- Inductive learning
  - generate rules from raw data
  - similarity-based learning, case-based reasoning
- Deductive learning
  - general rules are used to determine specific facts
  - theorem proving
- Augmenting knowledge-base is not a simple task
Reading

- S Haykin, Neural Networks: A Comprehensive Foundation, 2007 (Chapter 1).