

## Neural Networks / Vision

### Homework:

**Reading:** A. K. Jain et al.,  
 “Artificial Neural Networks: A Tutorial”  
 (link on class web site)

**Reading Questions and Exercises**  
 (link on class web site)

Note: Different assignments depending on your knowledge of neural networks

## Face recognition via Neural Networks

(From T. M. Mitchell, *Machine Learning*, Chapter 4)

- **Task:** classify camera images of various people in various poses.
- **Data:** Photos, varying:
  - Facial expression: *happy, sad, angry, neutral*
  - Direction person is facing: *left, right, straight ahead, up*
  - Wearing sunglasses?: *yes, no*

Within these, variation in background, clothes, position of face for a given person.



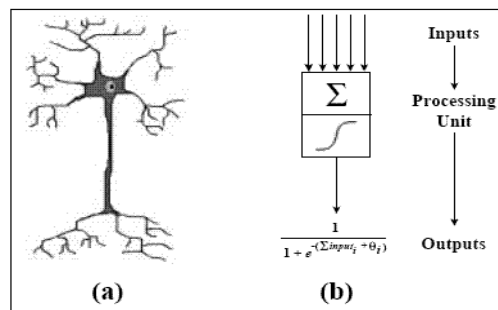
an2i\_left\_angry\_open\_4



an2i\_right\_sad\_sunglasses\_4



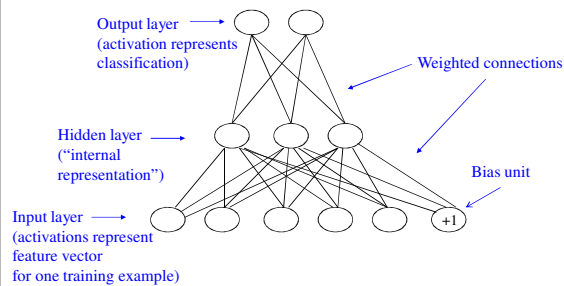
glickman\_left\_angry\_open\_4



biological neuron

artificial neuron

## A two-layer feed-forward neural network



## Training feed-forward neural networks

Assume two-layer network with random initial weights (small positive and negative values):

I. For each training example:

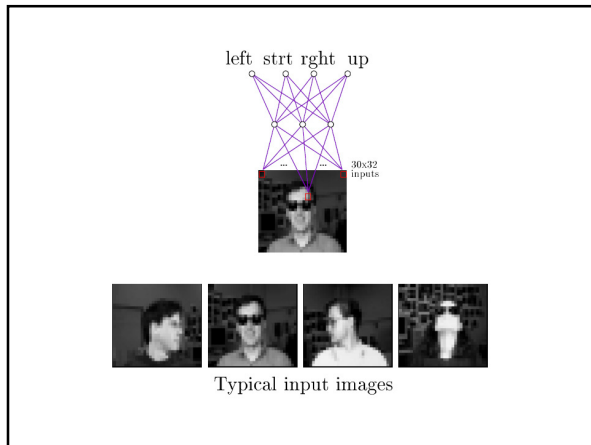
1. Present input to the input layer.
2. Forward propagate the activations times the weights to each node in the hidden layer.

3. Forward propagate the activations times weights from the hidden layer to the output layer.
4. At each output unit, determine the error  $E$ .
5. Run the back-propagation algorithm to update all weights in the network.
  - **Learning rate:** Determines magnitude of weight updates
  - **Momentum:** Determines magnitude of influence of previous weight change (to keep changes going in same direction)

II. Repeat (I) for a given number of “epochs” or until accuracy on training or test data is acceptable.

## Design Choices

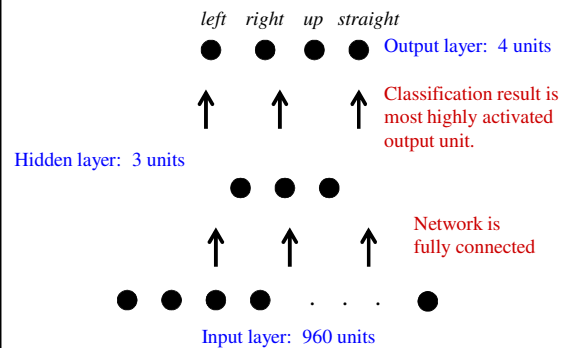
- Input encoding
- Output encoding
- Number of hidden units
- Learning rate
- Momentum



- Preprocessing of photo:
  - Create 30x32 coarse resolution version of 120x128 image
  - This makes size of neural network more manageable
- Input to neural network:
  - Photo is encoded as 30x32 = 960 pixel intensity values, scaled to be in [0,1]
  - One input unit per pixel
- Output units:
  - Encode classification of input photo

- Possible target functions for neural network:
  - Direction person is facing
  - Identity of person
  - Gender of person
  - Facial expression
  - etc.
- As an example, consider target of “direction person is facing”.

## Network architecture



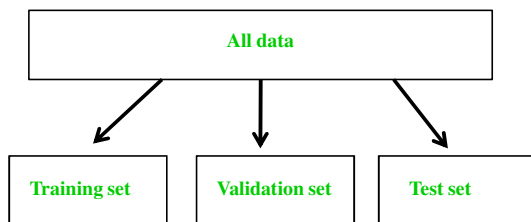
## Target function

- Target function is:
  - Output unit should have activation 0.9 if it corresponds to correct classification
  - Otherwise output unit should have activation 0.1
- Use these values instead of 1 and 0, since sigmoid units can't produce 1 and 0 activation.

## Other parameters

- Learning rate  $\eta = 0.3$
- Momentum  $\alpha = 0.3$
- If these are set too high, training fails to converge on network with acceptable error over training set.
- If these are set too low, training takes much longer.

## Training



## Training

- For maximum of  $M$  epochs:
  - For each training example
    - Input photo to network
    - Propagate activations to output units
    - Determine error in output units
    - Adjust weights using back-propagation algorithm
  - Test accuracy of network on validation set. If accuracy is acceptable, stop algorithm, and run resulting network on test set.

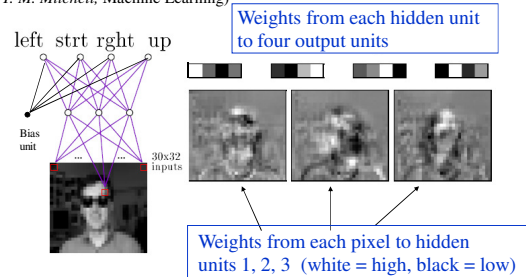
- [Demo of code](#)

Epoch	delta	trainperf	trainerr	t1perf	t1err	t2perf	t2err
0	0.0	50.9025	0.0972029	46.7626	0.103391	0.0	0.0
1	29.4377	50.9025	0.07842	47.482	0.0805035	0.0	0.0
2	30.6895	59.2058	0.0737351	53.2374	0.0763508	0.0	0.0
3	31.2238	67.87	0.0665212	63.3094	0.0704276	0.0	0.0
4	30.8616	76.8953	0.0578792	66.1871	0.0638382	0.0	0.0
5	29.4834	80.5054	0.0506615	71.223	0.0588367	0.0	0.0
6	27.4804	83.7545	0.0452837	75.5396	0.0551372	0.0	0.0
7	25.418	85.1986	0.0412303	74.8201	0.0522873	0.0	0.0
8	23.5952	85.5596	0.0380707	76.9784	0.0501809	0.0	0.0
9	21.9776	85.9206	0.035415	76.9784	0.0485298	0.0	0.0
10	20.3628	86.2816	0.0330951	78.4173	0.0469859	0.0	0.0

## Understanding weight values

After 100 epochs of training.

(From T. M. Mitchell, Machine Learning)



Hidden unit 2 has high positive weights from right side of face. If person is looking to the right, this will cause unit 2 to have high activation. Output unit *right* has high positive weight from hidden unit 2.