NOTES ON THE HUMAN BRAIN

Among the cells in the brain are those called neurons, which are communication cells that can signal one another via connections (synapses). Moreover, each synapse has a strength or weight that affects the degree of influence a signal coming from one cell has on the receiving cell. The neurons and the synapses between them thus form an enormously large directed and weighted graph. How large? Well, there are at least 10^{11} (100,000,000,000 or 100 billion) neurons in the human brain, and on the order of 10^{15} synapses. That's big!!!

Aside from it's incredible size, why is such a graph interesting? (i) it's our brain, and supposedly the basis for who we are; (ii) it changes over time: old neurons die, new ones grow, new connections form, weights change with experience; (iii) it is an active structure, sending information around inside itself, rather than a more typical graph that is simply a passive data storage that a separate algorithm acts upon; (iv) is is massively parallel: activity goes on in billions of neurons at the same time, unlike a typical graph algorithm that travels along only one edge (between two vertices) at a time.

Thinking about the brain this way has stimulated research in computer science, to pursue computational models of massive parallel processing. There are two rather distinct areas that have come about: (i) artificial neural networks, and (ii) abstract parallel processing machines. The former (ANNs) are more closely patterned on actual neuronal behaviors, and often are used as models of the brain, as well as in artificial intelligence, with a special focus on learning or training based on experience. The latter provide designs for parallel computer architectures aimed at achieving much greater processing speed than conventional computers have. We will take a (very brief) look at artificial neural networks, after an equally brief look at the brain.

Most of our neurons and synapses are contained in the "neocortex" (present in mammals); it wraps around a smaller and evolutionarily older part of the brain sometimes called the "smell brain", "crocodile brain", or "reptile brain". In additon to the many neurons, there are other types of cells (some in even larger numbers), as well as fluids (blood, etc) and much more.

Much of what is known about brain function (what it does) comes from case studies of brain injuries (wartime is great for neuroscience). When a small bit of brain is destroyed, the patient's behavior may change, and that is a clue to what that part might do (but only a clue). Often an injury causes bleeding, which then may in turn cause other problems if the bleeding continues: pressure can build up where the blood accumulates, and the pressure can damage neurons in that region.

The rough outline of the human brain seen from the left is:

```
       -----------------
      /      */#         )
f    (    F  */#  P     / \
r     \_____*/#_________/   \
o          _-        (  O  )
n         /     T     \   /
t        (_____________\_/
                 \    C  )
                  \   __)
                   \ \
```

Fig 1

where F = frontal lobe, P = parietal lobe, T = temporal lobe, and O = occipital lobe. The lower tip of the smell brain is just visible below the temporal lobe, where it joins the spinal cord; also visible is a bit of the cerebellum, C. The four lobes shown form the neocortex; inside, at the top of the spinal cord (not shown) are the older parts of the brain (hidden behind the temporal lobe). The occipital lobe is also known as V1: the primary visual cortex; information from the retinas (the eyes sit in the gap under F to the left of T above) passes via the optic nerve to V1 (doing all sorts of complicated things along the way).

From the top:

```
           --------------------
          /                    \
         |                      \
f        |                      /
r         \____________________/
o          _______XXXXX___________
n         /                    \
t        |                      \
         |                      /
          \____________________/
```

Fig 2

The top view shows that the brain (neocortex) comes in two very similar "hemispheres", left and right, each of which look like Fig 1 above when seen from the side. Thus there are two frontal lobes, two parietal lobes, etc. The two hemispheres are connected by a thick band of axon fibers XXXXX called the corpus callosum.

The strip $***$ along the back edge of F, and the strip ### along the front edge of P, are called the motor strip and the somatosensory strip, resp. The motor strip consists of neurons whose signals proceed down the spinal cord to muscles, causing them to contract; the frontal lobes (left and right) are thought to be involved in planning, and thus it seems not unreasonable that motor commands (enacting a plan) would emanate from there. The somatosensory strip receives sensory signals traveling from the body up through the spinal cord and smell brain and other inner structures and into the parietal lobe.

A striking feature is that of the "crossed" brain: all of the motor and sensory signals that travel to and from the brain via the spinal cord cross over to the opposite side. That is, the left half of the body (below the neck) is connected (both for sensory and for motor signals) to the right side of the brain, and vice versa.

The two eyeballs are situated just under the left and right frontal lobes. The "crossing" situation for vision is more complicated than for the motor and somatosensory strips just mentioned. Each eye (at the back, or retina) sends via its light-sensitive neurons (rods and cones) signals to either the right or left occipital lobe, depending not on which eye but rather on whether the light comes from the right or left of center of gaze. That is, half of the retina in each eye projects to the left occipital lobe, and half to the right occipital lobe. Specifically, light from the right of center falls on the left half of the retina in each eye, and then signals are sent from there to the left side of the brain, and similarly for the right. (However, the corpus callosum transmits this information from each side also to the other side.) Destruction of, say, the left occipital lobe, results in complete loss of visual awareness of the right visual field, and vice versa.

It is known that the occipital lobe performs only the "early" visual processing, such as edge orientation, and that this information in turn is passed to other areas such as the temporal lobe where a presumed (but ill-understood) process of integration occurs, affording high-level determination of what is seen (eg, a house or a face). The temporal lobe is also involved in memory and in processing of auditory information. The parietal lobe is thought to process highly abstract ideas such as mathematics.

Overall, however, although many volumes of detailed information about the brain are well established, *how* it manages to do what it does is still in many respects uncertain. Even vision, the most heavily studied brain function, remains cloudy in at least one fundamental respect: at what point, and as part of what process, does visual awareness (actual subjective experience of seeing) occur? Not only is this not known, but no one has even managed to formulate a clear guess as to what it could be. The easiest notion to form (and then reject!) is that some sort of image is created in the brain, a bit like a TV set (but then "who" is looking at it?).

Now let us return to individual neurons. A neuron is a highly specialized cell, with the usual cell body with its nucleus as well as a long "axon" that acts a bit like a current-conducting wire. The axon usually splits into many "tips" that can come close to other neurons.

```
                                           -------
                                          |       |
                                  -----   |       =====
 ----------                      /        |_______|
|          |              /  __/       ^      another neuron
|          | signal -->  /  /       synapse
|          ===============-----
|          |    axon     \   \
|__________|              |   \__

 cell body
```

When the proper electro-chemical conditions occur in the cell body, an electrical current ("action potential") is initiated there, which travels from the cell body all the way along the axon to the axonal tips where it causes molecules known as neurotransmitters to be released. If there is another neuron close enough, some of the neurotransmitters will come into contact with it. This close proximity that allows such contact is called a synapse. A synapse can be such that, when the neurotransmitter makes contact, the contacted cell becomes more likely to fire (excitatory synapse) or less likely (inhibitory synapse).

The cerebellum (C, in Fig 1) consists mostly of inhibitory synapses, which seems not unreasonable given that its apparent function is to provide fine motor control (as in piano-playing or reaching for something); damage to the cerebellum leads to loss of this control so that motions tend to be exaggerated as in overreaching.

One neuron, on average, will be able to send signals directly to (ie, have axons synapsing with) on the order of 5000 others.

(ARTIFICIAL) NEURAL NETWORKS

Neural networks are abstract mathematical models of interconnected neuronal behavior. Perhaps the first such model was that of the McCullough-Pitts artificial neuron in 1949. This postulates a unit of processing (a cell body and an axon) with characteristics indicated:

```
      --------
a1---|Wj1 |   |
a2---|Wj2 |   |
a3---|Wj3 |Hj |--------------- aj
...  |... |   |
an---|Wjn |   |
      --------
       unit j
```

Here the ai are incoming signals (O or 1) that arrive at the processing unit j, where each is multiplied by a weight Wij and the result summed: $SUM = a_i W_{j_1} + a_2 W_{j_2} + ... + a_n W_{j_n}$. This sum is then compared to the threshhold value H_j for the unit. Finally, the output signal a_j (which is sometimes called the activation level of the unit) is defined as

```
      { 1 if SUM >= Hj
a_j = {
      { 0 otherwise
```

More precisely,

```
            { 1 if SUM(t) >= Hj
a_j(t+1) = {
            { 0 otherwise
```

where $SUM(t) = a_i(t)W_{j_1} + a_2(t)W_{j_2} + ... + a - n(t)W_{j_n}$.

A neat mathematical notation for this is $a_j(t+1) = \Theta(SUM(t) - H_j))$ where $\Theta(x) = 1$ if $x \geq 0$, and 0 otherwise (the so-called Heaviside function). Using the vector dot-product we can write this even more compactly as $a_j(t+1) = \Theta[a(t).W_j - H_j]$.

That is, the incoming signals ai vary dynamically at each time step, and the output signal aj is recomputed and sent one time-step after the incoming signals arrive. The weights Wij can be any real numbers; a negative weight can reduce the overall incoming contribution to SUM, and corresponds to an inhibitory synapse.

By a (neural) network is meant a collection of such units that are connected to one another via signal wires. An output wire such as aj above can split and connect to many units. Note that in principle aj(t) can be an input to unit j and thus can influence aj(t+1). A network that allows this is called "recurrent"; one prominent example is the "complete" network in which *every* pair of units is connected by a wire.

Here is a simple example of a (recurrent) network:

```
          --------
 -->-----| -1 | 0 |-->-
|         --------     |
|                      |
 --------<-------------
```

This has just one unit which feeds back to itself. It "blinks" since whatever the incoming signal is on the left (say on, or 1), it produces the opposite (eg off, or 0) on the right one step later, but then that (eg 0) becomes the new incoming signal, etc. So the output signal perpetually changes back and forth: 0,1,0,1,0,...

In addition to recurrent networks, there are so-called feedforward networks, in which each neuron is used only once in a given computation, as signals pass through it (from left to right) and then on to another "layer" of neurons:

```
layer 0   1   2   3
      |   |   |   |
      V   V   V   V

  ----o---o
       \   \
        \   \
         \   \
  ----o---o---o---o----
         /
        /
       /
  ----o---o---o---o----
       \
        \
         \
  ----o---o---o---o----

         ^^^^^^
  input  hidden   output
  layer  layers   layer
```

In the diagram, layer 0 is the input layer, layers 1 and 2 are "hidden" layers, and layer 3 is the output layer.

Here is a feedforward network that acts like an AND-gate:

```
      -------
p----|   |   |        The only way for the input contribution to
     | 1 |   |        reach the threshhold of 2 is for both p and
     |___| 2 |____    q to be "on", ie to be 1's.  If we interpet
     |   |   |        1 as true (and 0 as false) then this is an
     | 1 |   |        AND gate: it fires (produces an output of 1)
q----|   |   |        iff both inputs are true.
      -------
```

Notice that there are no hidden layers here: just an input layer (p and q) and an output layer, for a total of three units. Also one can easily and similarly create OR-gates, NOT-gates, and many other logic gates.

But it turns out that to make an XOR-gate (true iff exactly one of itw two inputs is true) one must have at least one hidden layer, as in:

```
      -------
p----|   |   |                  ----------
     | 1 |   |                 |     |    |
     |___| 1 |_________________|     |    |
     |   |   |                 |  1  |    |
     |-1 |   |                 |     |    |
q----|   |   |                 |     |    |
      -------                  |-----|    |
      -------                  |     |    |
p----|   |   |                 |     |    |
     |-1 |   |                 |     |    |
     |___| 1 |_________________|  1  |    |
     |   |   |                 |     |    |
     | 1 |   |                 |     |    |
q----|   |   |                 |     |    |
      -------                  |-----| 1  |------ P XOR q
      -------                  |     |    |
p----|   |   |                 |     |    |
     | 1 |   |                 |     |    |
     |___| 2 |_________________| -1  |    |
     |   |   |                 |     |    |
     | 1 |   |                 |     |    |
q----|   |   |                 |     |    |
      -------                  |-----|    |
      -------                  |     |    |
p----|   |   |                 |     |    |
     |-1 |   |                 | -1  |    |
     |___| 0 |_________________|     |    |
     |   |   |                 |     |    |
     |-1 |   |                 |_____|____|
q----|   |   |
      -------
```

Here there are only two input layer units, p and q, as before, but for ease of drawing I have shown their connections to all four hidden units without showing the crossing wires that would be needed if they are all lying on the same plane. The top two hidden units fire if either p is true and q false (top unit) or vice versa (second unit). The bottom two fire if either both are true (third unit) or neither (last unit). This exactly one of the four hidden units can fire for any given input values. [In fact, as one student pointed out, the bottom two units and their connections to the output unit can be removed altogether and the simpler network (only two hidden units) still computes XOR.] With enough hidden units in a feedfoward netwrok, it is possible to compute any computable function.

For feedforward networks there is a famous and much-used algorithm, "error backpropagation", that allows the networks synaptic weights Wij to be adjusted to fit with "training data" so that the outputs are the desired ones for given inputs; and after that the network often tends to exhibit the desired input-ouput relationship even for new data. For instance, suppose a feedforward network is trained on 100 instances of the handwritten letter "E" and another 100 that are not E's. Let us further suppose that there are, say, 625 input units (corresponding to a 25x25 grid of pixels) and two output units (corresponding to "E and "not-E"). Once its weights are adjusted by repeated runs in the error-backpropagation learning algorithm so that it correctly categorizes the 200 inputs, it can then be used with no further adjustment to distinguish new handwritten letters (as E or not-E). A more sophisticated network might be trained to distinguish all 26 letters, and upper and lower case as well. The rough idea behind backpropagation is to compare the actual output with the desired output (for a given input) and calculate a set of weight-alterations that will bring the ouput closer to what is wanted. The actual algorithm uses derivatives and requires a change in the basic formula for aj(t+1) so that instead of the Heaviside function, a differentiable substitute is used.

Recurrent networks can also be trained, and have found useful application when matching a new item with a set of stored ones to see which is the closest match. In this paradigm, there are no layers, and the entire network is used for input, as well as for output. Input values can be specified as activation values (0 or 1) on the connecting wires; and output values are also so specified, after the ensuing computations settle into a stable (unchanging) state.

An example is face-recognition: a complete network is trained on, say, 100 faces (photographs) so that, given any one of them as input (eg pixel data) it it simply remains in that state. Then a new photograph is presented, and the network goes through stages of processing as activation levels change until eventually each wire reaches a final activation (0 or 1) that no longer changes. It turns out that (if the training was done properly) the final state is exactly one of the original 100 stored images, and (typically) the closest match to the new image.

While feedforward nets can also be used for this purpose, complete networks can store far more information due to the very large number of connections. Viewed as storage devices, complete networks are said to implement a kind of

”associative memory.”