Machine Learning - Some Examples

Data with features

Machine M

$\{0, 1, \ldots, 9\}$

Bit maps of handwritten numerals

$16 \times 16$ matrix
Input

has correct output $y_i$

So want $y_i = M(x_i)$  \(\text{correct answer}\)

How build $M$?

Rule based expert system - deterministic, fixed

New idea
$M$ depends on parameters $\mathcal{W}$ and can be learned by adjusting $\mathcal{W}$.

Given $\vec{x}_1, \ldots, \vec{x}_n$ training data; $y_1, \ldots, y_n$ correct output.

1. Pick $\mathcal{W}_0$, initial parameters.
2. For $i = 1 \rightarrow n$
   - $\vec{x}_i \rightarrow M_{\mathcal{W}_i}(\vec{x}_i)$
   - adjust $\mathcal{W}_i$ to Shrink $|M_{\mathcal{W}_i}(\vec{x}_i) - y_i| - \text{error}_i$
3. End

- $M_{\mathcal{W}_n}$ then used on new data and see how it does.

- **Note**: Often multiple $\vec{x}_i$ are processed before adjusting $\mathcal{W}$. 
What form should the machine take?

- Popular and successful structure is neural net—many variants
- First version
- Explain later what motivates this choice.
$M_{m_2}(x) = \prod_{k=1}^{m_2} F_k \circ F_{k-1} \circ \ldots \circ F_1(x)$

each $F_i(x, A_i, \vec{b}_i) \quad A_L$ is matrix

$= \sigma \left( A_L \vec{x} + \vec{b}_i \right)$

activation function
\[ f(y) = y \]

\[ \nabla (y_1, y_2, \ldots, y_n) = (\nabla (y_1), \ldots, \nabla (y_n)) \]

Vectorized version

Deep feed forward neural net

\[ \text{many layers or functions } F \]