Notation

To ensure that everyone is up to speed on notation, let's review

- the notation (ML_Notation.ipynb) that we used in the "Classical Machine Learning" part of the intro course.
- additional notation (Intro_to_Neural_Networks.ipynb) used in the "Deep Learning" part of the intro course

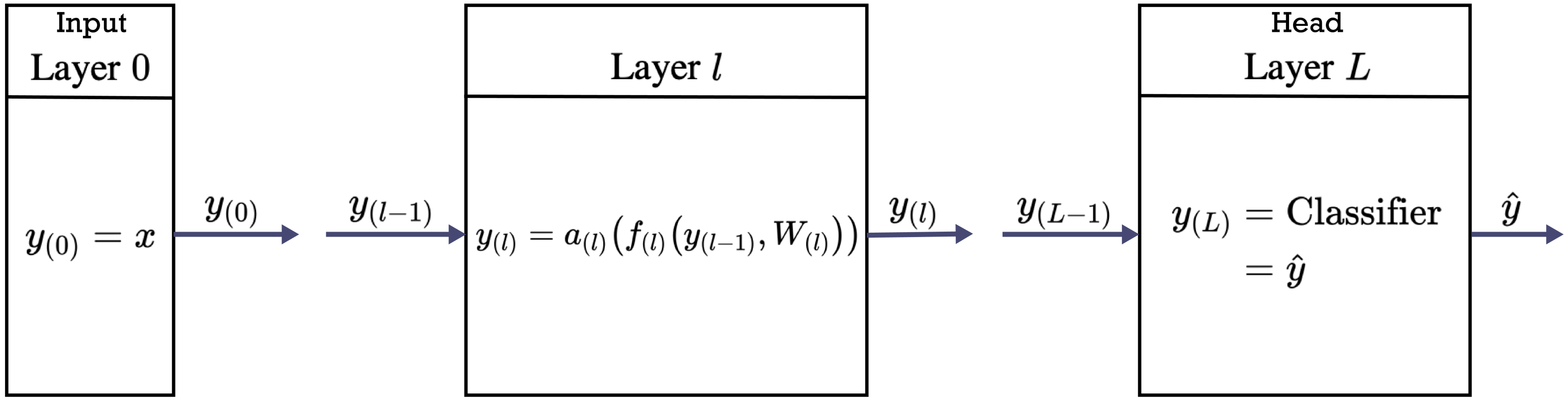
Representations

A path through a Neural Network can be viewed as a sequence of representation transformations

- transforming *raw features* $\mathbf{y}_{(0)} = \mathbf{x}$
- into *synthetic features* $\mathbf{y}_{(l)}$
  - varying with layer $1 \le l \le (L-1)$
- of increasing abstraction

Thus, the output anywhere along the path is an *alternate representation* of the input

Path through a Neural Network



Input
Layer 0
$y_{(0)} = x$
$y_{(0)}$
$y_{(l-1)}$
Layer l
$y_{(l)} = a_{(l)}\left(f_{(l)}\left(y_{(l-1)}, W_{(l)}\right)\right)$
$y_{(l)}$
$y_{(L-1)}$
Head
Layer L
$y_{(L)} = \text{Classifier}$
$= \hat{y}$
$\hat{y}$

Shallow features are less abstract: "syntax", "surface"

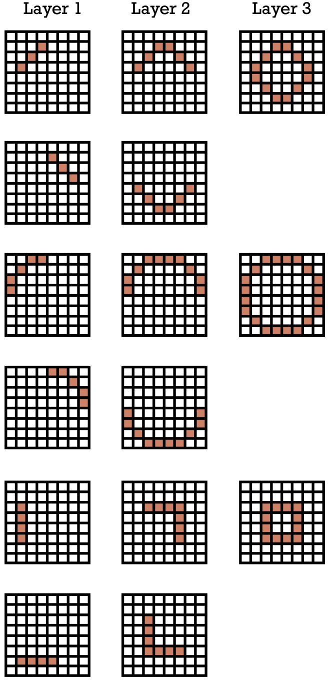
Deeper features are more abstract: "semantics", "concepts"

- We may even interpret the features as "pattern matching" regions or concepts in the raw feature space.

For example, in a CNN

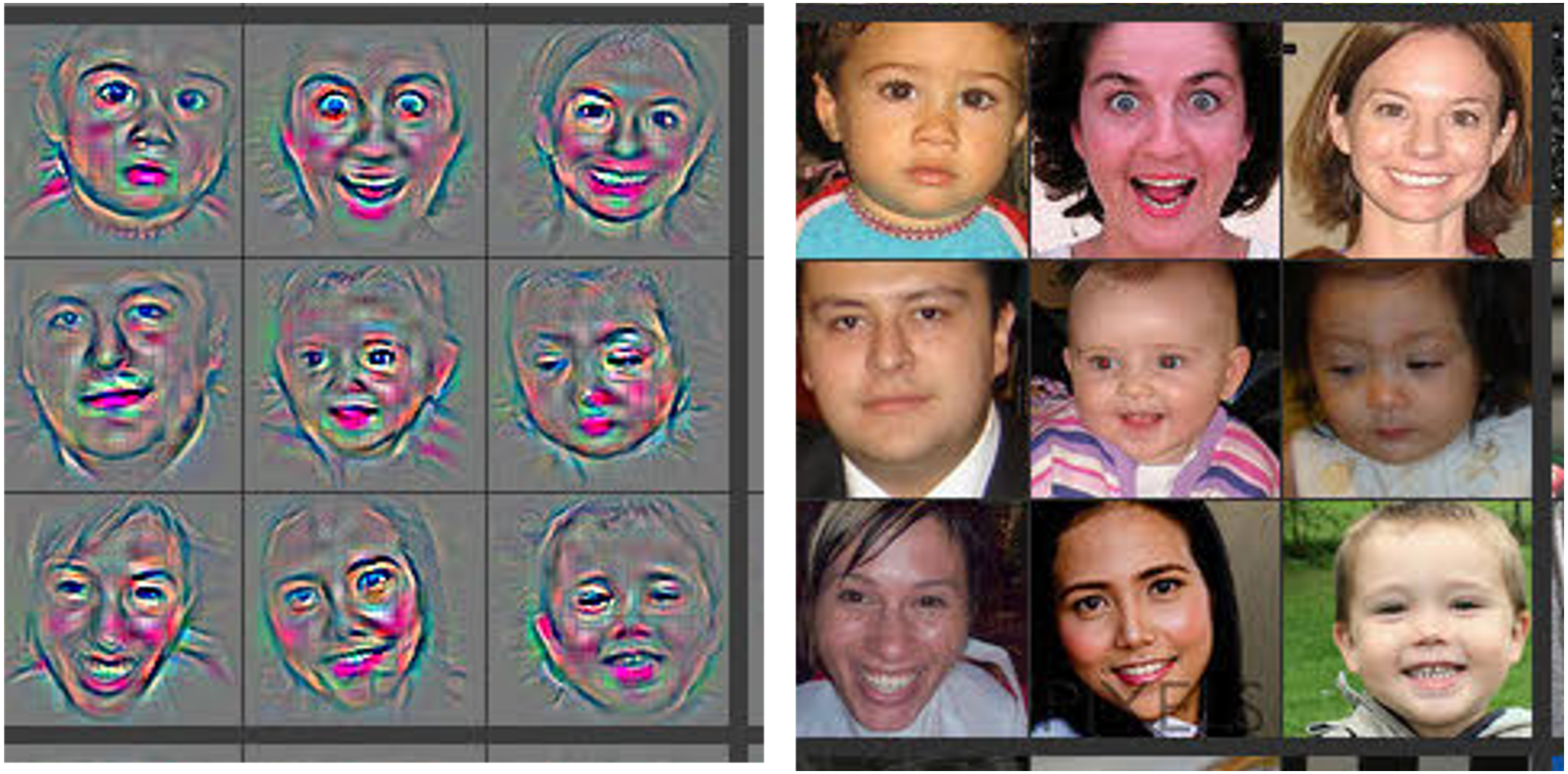
- shallow features are primitive shapes
- deeper features seem to recognize combinations of shallower features

Input features detected by layer



Layer 1
Layer 2
Layer 3

Saliency Maps and Corresponding Patches
Single Layer 5 Feature Map
On 9 Maximally Activating Input images



Layer 5 ? Feature Map (Row 11, col 1).

Attribution: [https://arxiv.org/abs/1311.2901 (https://arxiv.org/abs/1311.2901)](https://arxiv.org/abs/1311.2901)

In the simple architectures of the Intro course, we mostly ignored the intermediate representations

$$\mathbf{y}_{(l)} : \; 1 \le l \le (L-1)$$

The layers were referred to as "hidden" for a reason !

We will discover uses for intermediate representations and show how to build a "feature extractor" to obtain them from a given architecture.

In [2]:

```
print("Done")
```

Done