1 | Motivation

The logic behind CNNs is questionable. This is evident mostly in the presence of the pooling layer, which serves to add an element of invariance (i.e. if you wiggle the image around a bit, the prediction should be the same). However this does not actually address the problem of pose and perspective.

Capsule Networks are a solution to this that are modelled more similarly to the brain.

2 | Capsules

Capsule are organized in a surprisingly hierarchical manner. Capsules are clumps of neurons that have "children" and "parents", with another main difference being their output form. Capsules output vector representations of the activation as opposed to a single scalar. The magnitude of a capsule output represents the likelihood of a prediction, while the direction of it signifies properties of the input.

An element of nonlinearity is then introduced with the squashing function, and then the output is passed upwards to its parent.

The main idea of the vector output is to avoid the problem with pooling and CNNs where changes in image properties like perspective affect likelihood of prediction. When pose changes in the input of a CapsNet, the output vector's magnitude stays the same but its direction spins around with the changing pose. Because of the more complex representation the information and prediction are preserved.

3 | The Math

A capsule takes in a vector from its child capsule and runs an affine transform on it: $\widehat{\mathbf{u}}_{j|i} = \mathbf{W}_{ij}\mathbf{u}_i$. This is then brought into the traditional part of a layer (normally the $w_i x_i + b$ part) $\mathbf{s}_j = \sum_i c_{ij}\widehat{\mathbf{u}}_{j|i}$. Finally, they apply the nonlinearity squashing function $\frac{||\mathbf{s}_j||^2}{1+||\mathbf{s}_j||^2}\frac{\mathbf{s}_j}{||\mathbf{s}||_j}$.

4 | Structure

A capsule network begins with normal convolutional layers that feed into the capsules.