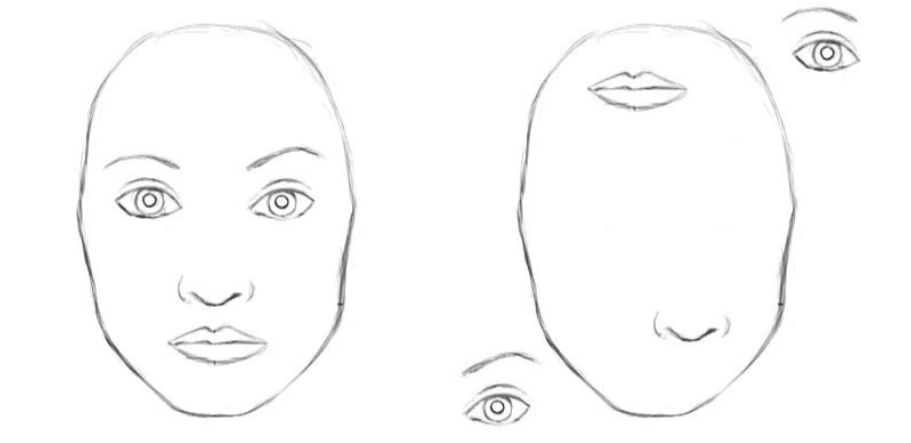
**Capsule Networks**

**Why is Capsule Networks becoming popular?**

Convolutional neural networks (CNNs) has seen a lot of incremental developments over the years and has hugely contributed to the success in the field of deep learning. They have an extensive learning capacity and has been a very suitable method for a lot of image classification problems. However, they have their limits and a few fundamental drawbacks. CNNs recognize an object or pattern in an image by detecting its features as a group of pixel values. This conveys that a mere existence of an object in an image is a very strong indicator of its presence in the image and this information is used to make the final predictions from the network model. This is where the flaw lies in the system as this method solely cares about the presence of an object and ignores the spatial relations and the orientation of the object in an image. A more intuitive visualization of the problem is depicted below. From the below images, a CNN can recognize both the images on the left and right to be a face. However, this is not the expected accurate prediction for the image on the right.



The CNN architecture fails to map the relationship between the different parts of the face (eyes, nose and mouth) in the prediction process. The most important constituent to this problem is in the usage of Max Pooling layers. A lot of valuable spatial information is lost in the process of max pooling layers as only the most active neurons are chosen to move further to the next layer. To overcome this issue, Hinton Et al [1] proposed a process called “routing by agreement” using capsule networks or CapsNet.

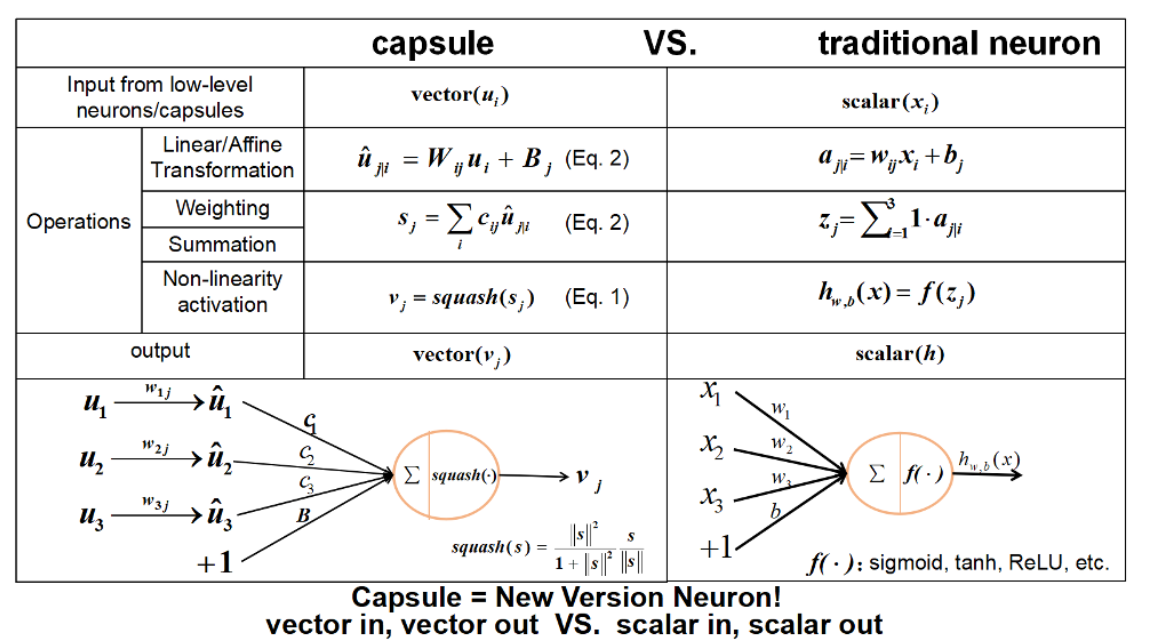
**What are Capsule Networks and what is its high-level architecture?**

A CapsNet is made up of capsules which are a group of neurons that learns to recognize an object with some sort of geometric information that is related to the position and orientation of the object in the image. The output from the capsules is a vector, with its length representing the probability that an object is present and its orientation (for example, in 8D space) representing the object’s pose parameters such as the rotation and the precise position of the object.

Like a regular CNN, the CapsNet architecture has multiple layers of which the lowest layer is referred to as primary capsule layers. The primary capsules receive a small region of the image where it tries to locate the presence and pose of a pattern. Capsules in the higher layers referred to as routing capsules try to recognize larger and complex objects, such as ships and boats. The “routing by agreement” process comes into play between these layers, where the primary capsule layer tries to predict the outcome of the capsules in the routing layers and activates the capsules in these higher layers only if the predictions agree. This iterative process of agreement-detection and routing update has achieved a considerably better performance that the regular CNNs with complex images containing overlapping patters of data.

**How does capsules work?**

While a single neuron in CNN receives a scalar input, multiplies them by scalar weights and returns a scalar output from the activation function, a single capsule implements vector forms of the above steps. The following figure gives a summary of the operations involved with a capsule and traditional neuron.



From the left-hand side of the image, we can see that there are 4 different operations that happen between receiving the input and producing output of a capsule.

**Step1:** At first, the input received is of vector form from 3 of the previous capsules. The lengths of these vector represent the probabilities that the intended object is detected by the lower level capsules. The direction of these vectors represents the object’s pose parameters. These are multiplied by weight matrices that represent the spatial relationship between lower level features such as nose, eyes etc. and higher-level features such as the face. For example, W1j would be to associate the relationship between the eyes and the face, W2j for the relationship between mouth and the face and so on.

**Step2:** Multiplying these weight vectors, results in a prediction that helps detecting the position of the higher-level features. For example, u1hat represents the prediction of the position the face could be detected with respect to the eyes and u2hat predicts the position of the face with respect to the mouth. These predictions are multiplied with a scalar that will help decide to which higher level capsule should the output be routed to. Unlike the scalar weights that are learnt using backpropagation, the scalar in the capsule networks is based on the dynamic routing algorithm as proposed by Hinton which involves the iterative process of agreement-detection and routing update process as mentioned before.

**Step3:** Once this is done, the sum of the vectors is calculated.

**Step4:** Finally, a novel squashing function that takes in a vector and squashes it to have a length between 0 and 1, leaving the orientation unchanged. This indicates the output length of the vector is a probability again of a feature being detected by the capsule.

Based on this iterative process, the capsules learn to associate the lower level features with respect to the higher-level features, by maintain the spatial relationships between the objects in an image.

**References**

<http://papers.nips.cc/paper/6975-dynamic-routing-between-capsules.pdf>

<https://ieeexplore.ieee.org/document/8451379/references#references>

<https://medium.com/ai%C2%B3-theory-practice-business/understanding-hintons-capsule-networks-part-i-intuition-b4b559d1159b>

<https://towardsdatascience.com/capsule-networks-the-new-deep-learning-network-bd917e6818e8>

<https://towardsdatascience.com/capsule-neural-networks-part-2-what-is-a-capsule-846d5418929f>

<https://www.oreilly.com/content/introducing-capsule-networks/>

<https://github.com/naturomics/CapsNet-Tensorflow>