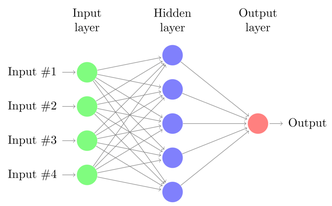
Residual Capsule Network

Can Zhong, Lingfeng Wang
Dept. of Computer Science, CEAS, University of Wisconsin - Milwaukee

Introduction

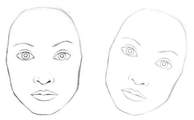
Artificial intelligence is the ability for an Agent to make practical and meaningful decisions in a dynamic or static environment. Machine learning is a means to accomplish artificial intelligence decision making through the implementations of algorithms which can find numerical relationships between data points without being explicitly programmed.

Due to the accumulation of massive amounts of data and the emergence of efficient computational hardware, deep learning has been a popular framework which has gain traction in research and applications in tasks such as classification, decision-making, and prediction.



CapNetv1 – Dynamic Routing

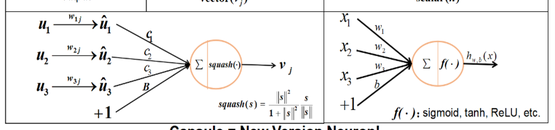
The most prominent architecture in use are convolutional neural networks which learn shared "concepts" throughout the data point by recognizing trends in a smaller space. CNNs excel at understanding the hierarchical properties of information yet one of its challenges is keeping perspective independent from the learned features.



Equivariance
Same Thing, Different Take

Capsule networks are an expansion of CNN which consider perspective or pose independent of activations. This architecture can be done with less parameters because the network does not need to learn different feature map for varying i.e. angles.

		capsule	VS.	traditional neuron
Input from low-level neurons/capsules		$\mathbf{vector}(\boldsymbol{u}_i)$		$\mathbf{scalar}(x_i)$
Operations	Linear/Affine Transformation	$\hat{\boldsymbol{u}}_{j\mid i} = \boldsymbol{W}_{ij}\boldsymbol{u}_i + \boldsymbol{B}_j$ (Eq. 2)		$a_{ji} = w_{ij}x_i + b_j$
	Weighting	$s_j = \sum_i c_{ij}\hat{\boldsymbol{u}}_{j\mid i}$ (Eq. 2)		$z_j = \sum_{i=1}^{3} 1 \cdot a_{ji}$
	Summation			
	Non-linearity activation	$v_j = squash(s_j)$ (Eq. 1)		$h_{w,b}(x) = f(z_j)$
output		$\mathbf{vector}(\boldsymbol{v}_j)$		$\mathbf{scalar}(h)$



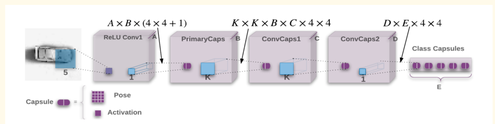
Capsule = New Version Neuron!
vector in, vector out VS. scalar in, scalar out

$$\mathbf{v}_j = \frac{||\mathbf{s}_j||^2}{1 + ||\mathbf{s}_j||^2} \frac{\mathbf{s}_j}{||\mathbf{s}_j||}$$

Methods

- To fully appreciate the intuitions behind neural networks, it was necessary for me to understand the fundamental concepts first. This includes the history, algorithms, architectures, statistics and calculus.
- To explore the recent development in various fields of neural networks, research papers were read and annotated.
- From these insights, 4 models were modified and then trained with Nvidia Tesla K80 and V100 GPUs on the MNIST dataset with good results. MNIST test results were good, a best of 98.82 %.
- Further experiments were trained and tested on a more complex dataset, CIFAR10 with the same models achieving a staggering low of 66.92 %. A conceptually-comparable network (DCNET) achieved 82.68 % with 16 million parameters. Further improved my model and re-ran. Reached a respectful 76.12 % while using only 4.5 million parameters.
- Results from CIFAR10 did not meet benchmarks but were promising. To beat the state of the art, I attempted to design an architecture which uses ODE-based functions for training and testing.

CapNetv2 – EM Routing



Procedure 1 Routing algorithm returns **activation** and **pose** of the capsules in layer $L+1$ given the **activations** and **votes** of capsules in layer L. V_{ij}^h is the h^{th} dimension of the vote from capsule i with activation a_i in layer L to capsule j in layer $L+1$. β_a, β_u are learned discriminatively and the inverse temperature λ increases at each iteration with a fixed schedule.

procedure EM ROUTING($\boldsymbol{a}, V$)
  $\forall i \in \Omega_L, j \in \Omega_{L+1}$: $R_{ij} \leftarrow 1/|\Omega_{L+1}|$
  for l iterations **do**
    $\forall j \in \Omega_{L+1}$: M-STEP($\boldsymbol{a}, R, V, j$)
    $\forall i \in \Omega_L$: E-STEP($\mu, \sigma, \boldsymbol{a}, V, i$)
  return $\boldsymbol{a}, M$

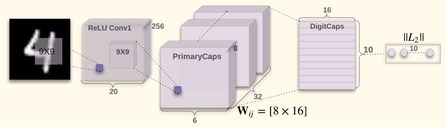
procedure M-STEP($\boldsymbol{a}, R, V, j$) ▷ for one higher-level capsule, j
  $\forall i \in \Omega_L$: $R_{ij} \leftarrow R_{ij} * \boldsymbol{a}_i$
  $\forall h$: $\mu_j^h \leftarrow \frac{\sum_i R_{ij} V_{ij}^h}{\sum_i R_{ij}}$
  $\forall h$: $(\sigma_j^h)^2 \leftarrow \frac{\sum_i R_{ij}(V_{ij}^h - \mu_j^h)^2}{\sum_i R_{ij}}$
  $cost^h \leftarrow (\beta_u + log(\sigma_j^h)) \sum_i R_{ij}$
  $a_j \leftarrow logistic(\lambda(\beta_a - \sum_h cost^h))$

procedure E-STEP($\mu, \sigma, \boldsymbol{a}, V, i$) ▷ for one lower-level capsule, i
  $\forall j \in \Omega_{L+1}$: $\boldsymbol{p}_j \leftarrow \frac{1}{\sqrt{\prod_h^H 2\pi(\sigma_j^h)^2}} \exp\left(-\sum_h^H \frac{(V_{ij}^h - \mu_j^h)^2}{2(\sigma_j^h)^2}\right)$
  $\forall j \in \Omega_{L+1}$: $\boldsymbol{R}_{ij} \leftarrow \frac{\boldsymbol{a}_j p_j}{\sum_{k \in \Omega_{L+1}} \boldsymbol{a}_k p_k}$

$$P_{i|j}^h = \frac{1}{\sqrt{2\pi(\sigma_j^h)^2}} \exp\left(-\frac{(V_{ij}^h - \mu_j^h)^2}{2(\sigma_j^h)^2}\right), \qquad ln(P_{i|j}^h) = -\frac{(V_{ij}^h - \mu_j^h)^2}{2(\sigma_j^h)^2} - ln(\sigma_j^h) - ln(2\pi)/2$$

$$a_j = logistic\left(\lambda\left(\beta_a - \beta_u \sum_i r_{ij} - \sum_h cost_j^h\right)\right)$$

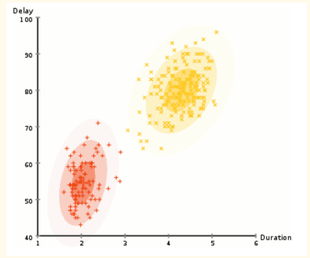
CapNetv1 - Dynamic Routing Architecture + Algorithm



Procedure 1 Routing algorithm.

procedure ROUTING($\hat{\mathbf{u}}_{j|i}, r, l$)
  for all capsule i in layer l and capsule j in layer $(l+1)$: $b_{ij} \leftarrow 0$.
  for r iterations **do**
    for all capsule i in layer l: $\mathbf{c}_i \leftarrow \mathtt{softmax}(\mathbf{b}_i)$ ▷ softmax computes Eq.
    for all capsule j in layer $(l+1)$: $\mathbf{s}_j \leftarrow \sum_i c_{ij}\hat{\mathbf{u}}_{j|i}$
    for all capsule j in layer $(l+1)$: $\mathbf{v}_j \leftarrow \mathtt{squash}(\mathbf{s}_j)$ ▷ squash computes Eq.
    for all capsule i in layer l and capsule j in layer $(l+1)$: $b_{ij} \leftarrow b_{ij} + \hat{\mathbf{u}}_{j|i}.\mathbf{v}_j$
  return $\mathbf{v}_j$

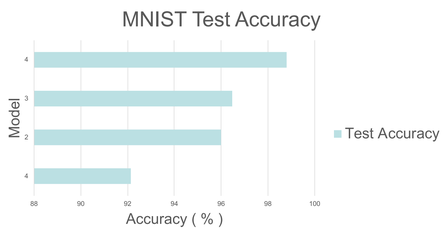
CapNetv2 – Expectation Maximization Clustering

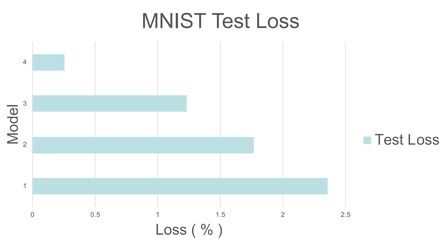


Experiment

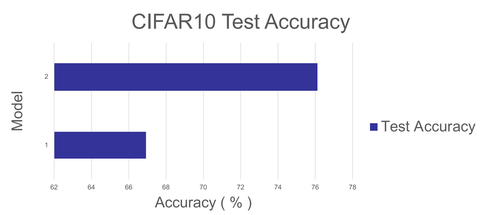
- In my experiment, I wanted to explore the hardware and computational ceilings that capsules face. Capsules are a relatively new concept and therefore development is in its infancy.
- I first wanted to test if the capsules inputs had any effects on the accuracy. I started through this route because I find the ability of a capsule to recognize patterns without having access to the whole very enticing.
- I hypothesized that if we changed what the input features were so that the input features described the datapoint more generic, then the classification accuracy would increase.
- I modified the networks in a way where the input to the primary capsules came from various networks. Baseline CNN, Residual, Residual with Cardinality, and an ODE-network.
- I found it quite slow to train my models as compared to other established architectures like CNN, and Dense Networks.
- I was limited to 1 GPU, but trained on the best Tesla V100.
- Residual Networks + Cardinality was an implementation which used more parameters than was necessary.
- ODE-network was the best option, but the problem with capsules is not how real data is encoded into the latent space, but rather how capsules normalize the flow from the latent space to the capsules.
- Feature endeavors include normalizing flow using capsules, and modifying the structure so that the state can use the last residual state. A problem with capsules is features learned earlier in the layer are more often than not to be forgotten in later layers.

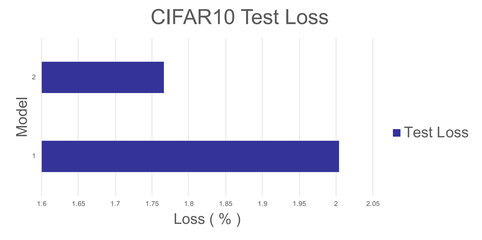
MNIST Results





CIFAR10 Results





Citations

Hinton, Geoffrey, Sabour, Sara, Frosst, Nicholas (2018). Matrix Capsules with EM Routing. *ICLR 2018.* 1-15.

Schmidhuber, Jurgen (2014). Deep Learning in Neural Networks: An Overview. *University of Lugano & SUPSI.* 1-88.

Xie, Saining, Girshick, Ross, Dollar, Piotr, Tu, Zhuowen, He, Kaiming (2017). Aggregated Residual Transformations for Deep Nueral Networks. *UC San Diego. Facebook AI Research.* 1-10.

Chen, Ricky, Rubanova, Yulia, Bettencourt, Jesse, Duvenaud, David (2018). Neural Ordinary Differential Equations. *University of Toronto, Vector Institute.* 1-19.

Sabour, Sara, Frost, Nicholass, Hinton, Geoffrey (2017). Dynamic Routing Between Capsules. *Cornell University, Computer Vision and Pattern Recognition.* 1-11.