Hyomin Seo

Title : Model Agnostic Meta Learning, Neural ODEs, Softmax Cross Entropy

1. Explain MAML

MAML

Model Agnostic Meta Learning

Meta Learning

Training a model on a variety of learning tasks, so that it can solve new learning tasks only using the minimum number of training samples

Model Agnostic Meta Learning

Reaches the Meta Learning objectives by providing a good initialization of the model's parameter.

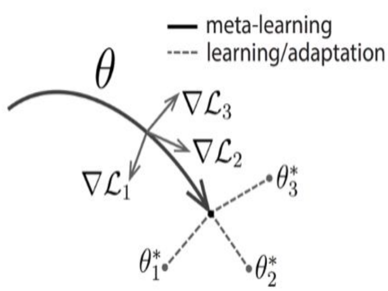
Given : model f_θ, *with parameter* θ

Given : Task Γ_i , *with association with* (D^i_{train}, D^i_{test}) ,

$$\theta_i' = \theta - \alpha \nabla_\theta \mathcal{L}_{T_i}(f_\theta)$$

Parameter update formula is presented :

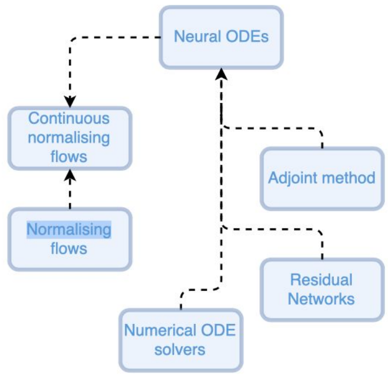
The presented update formula, with initial parameter, step hyper parameter, Delta of parameter, Loss (L) of task, and model, optimizes for an individual task, and finding an optimized θ^* will generalize such process to sets of data samples (then have two hyper parameters alpha/beta).



Visualization of the MAML approach. (Finn et al., 2017)

This particular MAML (like model agnostic on top of meta learning) is an improved version of meta learning : the algorithm can be combined with fully connected, convolution all, recurrent neural networks; therefore being able to adopt a variety of loss functions, either differential lesson or not. Easily out performs the general convolutional and recurrent model, showing its strongest strength on memory augmented network.

2. What are the main benefits and weaknesses of Neural ODEs

<table>
<tr><td>Neural ODE</td><td>

Neural ODEs are neural networks models which generalize standard layer to layer propagation to continuous depth models.
Neural ODEs take advantage of ‘Forward propagation = single step of discretion on ODE’ to construct and train efficient model, making continuous transform through generated vector field. </td></tr>
<tr><td>Benefits</td><td> The paper discusses 4 major benefits

Memory efficiency : Neural ODEs does not need backpropagation nor storing of any intermediate quantities, alleviating the heavy memory cost
Adaptive computation : Modern ODEs is adaptive for various evaluation strategy and rate of growth.
Scalable, invertible normalizing flows : Neural ODEs provide easier computation of change of variable formula, an advantage driven from continuous transformation instead of discrete steps.
Continuous time series models : Each time series is represented as a latent trajectory , letting the data to be naturally incorporated.
</td></tr>
<tr><td>Weakness</td><td> The paper discusses 4 major drawbacks

Mini Batching : Not that big of a problem except that mini batching is just less straightforward.
Uniqueness: It is a constrain - the differential equation must be uniformly LIpschitz continuous in z and in continuous t.
Setting tolerances: The error tolerance must be set for both forward and reverse pass (also not that big of a problem?)
Reconstructing forward trajectories : If the reconstructed trajectory diverges from original when the dynamics is ran backwards, it can result in some numerical error.
</td></tr>
</table>

3. Does magnet loss require any extra label information per example compared to softmax cross entropy

Magnet Loss

Loss function adopted in Distance Metric Learning

Magnet loss maintains an explicit model of the distribution of the different classes in representation space instead of penalizing any individual example ; Instead of working on individuals, pairs, or triplets of data points, magnet loss operates on entire regions of the embedding space that the data points inhabit. Magnet loss models the distributions of different classes in the embedding space and works to reduce the overlap between distributions.

Softmax function

Extended Logistic function for multiclass classification, used in multinomial logistic regression. Softmax function depends on all elements in of output score : can not be applied to individual output score.

Cross Entropy Loss

Cross entropy indicates the distance between what the model believes the output distribution should be, and what the original distribution really is ; alternative of squared error. It is greatly used as Loss function in neural networks (such as softmax function itself)

Cross Entropy Loss with Softmax :

Derived : $\frac{\delta L}{\delta o_i} = p_i - y_i$