Today I’m going to talk about something about learning a regularizer and getting insight of traditional optimization algorithms.

It has 3 parts, the first part is an introduction and the next two part is ideas from two papers

It is well known that the heart problem in inverse problem in image processing is to reconstruct an image x from y equals Ax plus n, where y is the observation, n is the noise. A is the transform matrix

For example the picture below shows different kinds of A and noise, and we can use methods to reconstruct it.

Generally we have two ways to solve this problem.

The first way is to solve an optimization problem with iterations. We want to solve this minimization problem with two terms. The first term is to evaluate the similarity between observation y and Ax, and the second term is to regularize x. lambda is an parameter to balance the two parts.

The regularizer phi is the prior knowledge of x. for example for an signal we want it to be sparse, for an image we want it to heve low-rank and maybe low total variation. So the phi can be l1 norm constraint or nuclear norm constraint or TV penalty.

And second way is to learn the relation between x and y by a neural network. For example by a fully connected network or convolutional neural network.

This two methods have its advantages and disadvantages. Iterative method can solve and inverse problem with the minimization problem mentioned above but have low performance relatively and it takes long time to deal with one image. And learning method have good performance but is designed for one specific problem and data set. And we do not use the transform matrix A in learning, but choose to learn it, which waste many efforts.

So we may want to combine this two methods.

The first idea is coming from the optimization algorithm ADMM. And we want to solve this minimization problem, first to write the augmented lagrangian function. And then apply the ADMM iteration. The update of u is just a least square problem, and so it’s easy to update u and z, but it’s hard to update x, cause it has a regularization here.

In fact the update of x is just a proximal operator on z-u, with respect to the regularizer. Here is the definition of proximal operator. I have introduced many kinds of regularizer before, but the best regularizer is just the indicator of all natural image IM,so the best update of x to project x to the natural image set M, by the definition of proximal operator.

So now we have the idea to train a classifier D to approximate IM, and based on it to train a projection function P to approximate the proximal operator. by adding noise to clean image x we get perturbed image v. and use the pairs to train D and P.

Also the out come of P can be feed to D to make it better, and we also absorb the idea from GAN, that is generative adversarial network. Notice that the structure of P is divided into two parts. The left part is the encoding part and the right part is the decoding part. We train an addition classifier Dl with the latent layer of P and experiments shows that it can avoid overfitting.

The second idea is from the gradient descent. We have this form of gradient descent method of iteration. And we can see it from another aspect. We denote the regularization part as r(x) here.

Eta is the stpesize.

Assume that capital R be the gradient of r(x), and assume capital R(x) be linear, R(x)=Rx, here R is a matrix. and we can compute the minimizer of the above problem.

We know that to compute the inverse of a matrix we can use a nuemann series and we can apply it on the minimizer x star, and truncate it to the first N+1 term t compute, call it x hat.

We have two ways to compute the sum of the first N+1 term.

We set X0= identity and have two kinds of iteration, and have two different forms of the sum.

And apply it to compute x hat, we set x0 = eta A transpose y and we have the following two kinds of iterations here.

Notice that the form of the first iteration is just the same as the form of the gradient decent method if r is linear.

So we treat it as a heuristic method to apply it to situations where R is not linear. Also, we use a network to train the regularizer r(x). and we have the following two methods. The notation here is different. But the structure is the same idea. The first is called unrolled optimization method and the second is called neumann network method. The second method is in fact inspired by the first method and neumann series so it’s called the neumann network.

Here is the reference, and that all for the presentation.

And for summary, to combine iteration and learning is to first using network to learn a classifier, a proximal operator, that’s all to learn the structure of the natural image set, which is a better prior knowledge. With this prior knowledge, we can have better iteration method.

Also, to combine this two can solve on a specific kind of data set, but we can deal with all kinds of problems, make full use of the transform matrix A and the data set. In some situation it is a better idea.

Thank you for listening.