Multiple task age estimation training

Training set

We sort all the images into groups by each person. And use data augmentation to make sure the number of images for each person are all the same.

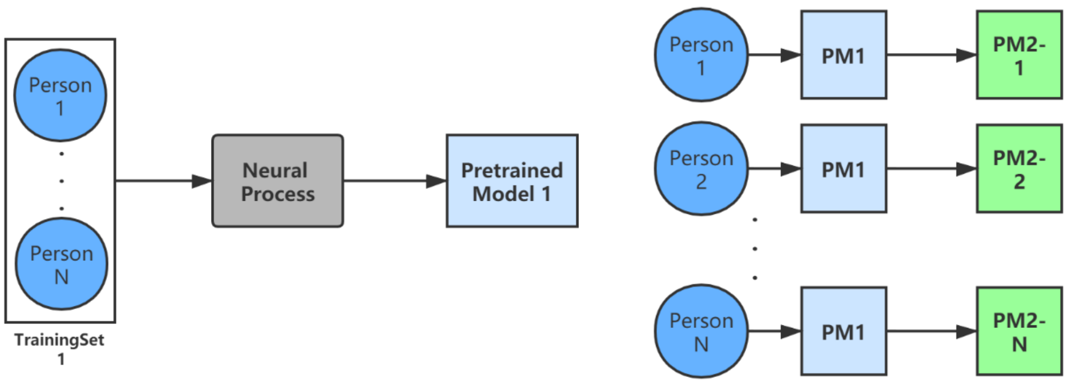
Each group of face images is denoted as $Person_i$.

Training Set 1 consists of N group of face images ($Person_{i...n}$)

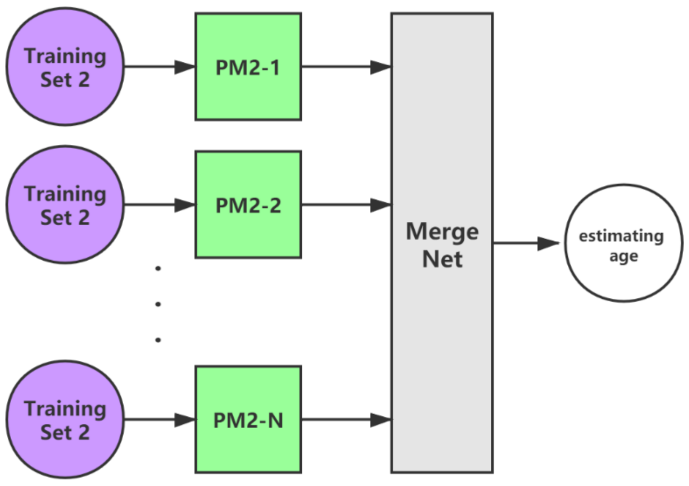
Training Set 2 consists of M groups of face images($Person_{n+1...n+m}$) which are all different from the components in Training Set 1.

Training Steps

Firstly, using the Training Set 1 to train the original Neural Process to get the Pretrained Model 1 which we denote as PM1. We consider each person as a different task in this process. In this round, we used the whole training set to make NP learn all tasks together. In this way, the Pretrained Model we get contains a general encoder which shares the same parameters in different tasks.

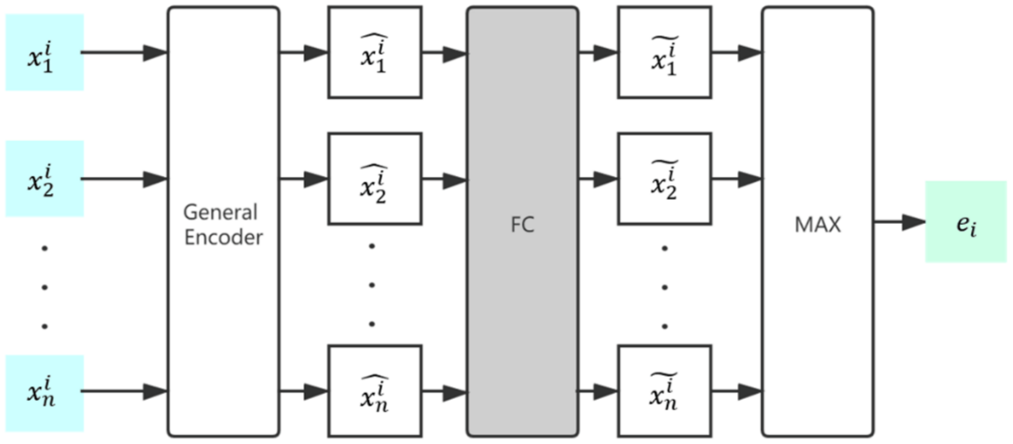


Then we train the PM1 separately with different $Person_i$ in Training Set 1 to get corresponding model PM2-i. In this process, the encoder part is fixed, we just use different task to train the decoder part. According to this, each PM2-i we get shares the same encoder part but possesses their unique decoder.



In this round, with all the pretrained model the weight will be calculated to determined the contribution of each decoder. This process can be divided into two parts.

a. Transform the representation

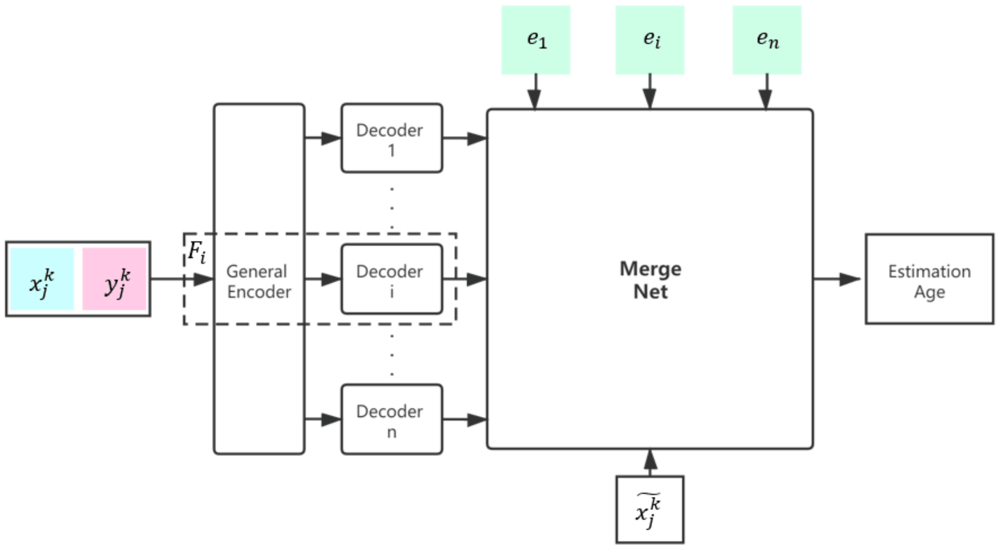


We denote the feature we get from ith person's jth image as x_j^i. Use the general encoder we trained in the first round we can get the representation from each feature. $\widehat{x_j^i} = E(x_j^i)$ Here $\widehat{x_j^i}$ is the original representation generated by the encoder. E(*) means the function of the general encoder.

The key point in this part is that we construct a fully connected layer which is defined as $\widetilde{x_j^i} = \sigma(W\widehat{x_j^i})$. The transformed representation is denoted as $\widetilde{x_j^i}$. W is the learnable parameters which we will train. σ is the activation function. In the implementation we choose ReLu here.

After it gets all the transformed representation of the person, the MAX operator will be conducted element-wise to get the object embedding $e_i = \max_j(\widetilde{x_j^i})$.

b. Calculate the weight and merge



In this part it needs to calculate the similarity between pth object embedding and the transformed representation of x_j^k . And we denote this as $s_{p,k,j}$

$$s_{p,k,j} = \frac{\exp\,(e_p^T \widetilde{x_j^k})}{\sum_{q \neq k} \exp\,(e_q^T \widetilde{x_j^k})}$$

We define the combination of the general encoder and decode i as the NP function F_i. With the input feature x we can get the prediction age y. Since it has n decode we need to do the merge. With the weight(similarity) the estimation age can be calculated :

$$\sum_{\substack{p=1 \\ p \neq k}}^{n} s_{p,k,j} F_p(x_j^k)$$

Actually, all the things we do here is to train the FC we defined. So the objective function to learn parameter is to minimize the difference between the face and our prediction.

$$\min_W \sum_{k=1}^{n} \sum_{j=1}^{n_k} \left(y_j^k - \sum_{\substack{p=1 \\ p \neq k}}^{n} s_{p,k,j}\, F_p(\widehat{x_j^k}) \right)$$

In the training stage, we will select and remove a single person x^i from the training set as the test set. If the training set has N person we will train the FC for at least N epochs. For each time, different people in the training set will be selected as the test object.