Matin jan well done. The codes were very clean.   
some points:

1) you might wanna use functions more. A piece of code for data preparation is exactly repeated in several files.

2) for neural network u also need to do cross validation to avoid overfitting. what u have done is only reporting the metric on validation set. In this way u should keep track of train/validation loss and decide on which epoch u want to stop training to avoid overfitting. If you want to control number of training iterations automatically u can use call backs in keras. It monitors the loss function on the validation set and saves the parameters of the best round of training on validation set. u may search about it urself.

If you search about it u see that auc roc score (which is the metric of this kaggle competition, cannot not be used in callback. u might find this link interesting to figure out why that is the case and how to overcome this issue.

https://stackoverflow.com/questions/41032551/how-to-compute-receiving-operating-characteristic-roc-and-auc-in-keras

3) embedding layers work best when u train them separately and feed a pretrained embedding layer to ur model. In this way ur model usually converges much faster. u might wanna search about it by googling "pretrained word embedding in keras"

 4) most importantly you can control the number of layers by checking if your model can overfit or not. In other words if your model improves both on ur training set and validation set as u increase the epochs it means that ur model is not complex enough and u can add a layer or increase the parameters of current layers. U should stop adding layers or parameters when ur training set loss decreases but validation set loss increases ( overfit point )

 5) its always good idea to visualize the loss function in the training process. In this way u can see the the rate of improvement and it helps u to choose a good learning rate and compare ur models.