some times like in the movie recommendation system the system recommends the movies on the base of people who have also watched
so while recommending this movies the system doesn't necessary to know what is the name of movie etc etc still it can recommend users some movies on the basis of what movies he have watched

here we are doing the analysis of movie rating and the user from the list of user in pandas dataframe which is available there

for merging the movie names with code of the dataframe we can use
ratings = ratings.merge(movies)
and after that we can get the movies along with the names code which it's gettinng passed

for loading the data with the dataloader api we can do it using
`dls = CollabDataLoaders.from_df(ratings, item_name = 'title',bs = 64)`

and we can use the `dls.show_batch()` for showing the batch

now we will need to create the number of users and the number of movies and for predicting them we need to have some number of factors which are gonna act as the weights for the model

for getting the latent factor we have to get the value by usign the dot product of instruction
looking up at the index can be represented as the matrix products
if we replace the indices with the one hot encoded vectoer it identical to looking up at the index

we are multiplying thoen hot encoded number to the user factors if we did this with the matrix multiplication is simiilar as the `dot_product`

`embedding` : embedding is matrix multiplication with the one hot encoding where you don't have to do one hot encoding cause it already know the gradients of one hot encode it's so far the efficient way for doing the matrix multiplication

`Creating the collaborative filtering from scratch`:

init method is the special method it can be used to initiate the function when you want to store some keywords while initialization

inheriting the properties of other class for using it in
we can do it like `class DotProduct(Module)` here we have inherited properties of Module class from python to the dot product class

the Module class is the pytorch class which is little bit modified by the fast ai
here we are defining the
self.user_factors = Embedding(n_users,n_factors)
here embedding layer is doing the matrix mutliplication with the dot product in more efficient way

while we call any function in pytorch it automatically calls the forward here we can use forward function help for doing the multiplication and the addition

it's the speical pytorch method this is where we are doing the actual computatoin method here we are passing the user id's and the movie id's

here we can pass the output as the sigmmoid range , pass the user bias etc thigns to normalize the model but it won't prevent overfitting . for preventing overfitting we have to do `regulization`

`Regulization`:

regularization is the set of techniques which allows us to use the models with lot of parameteres and train them for lot of period of time and penalize them effectively for preventing over fitting cause them to stop overfitting

⟵----------⟶ lesson 6 over ⟵---------⟶