Q1: Dimensionality Reduction using PCA and Reconstruction error

(a) Eigenvectors & Eigenvalues

In [1]:
```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
```

In [2]:
```
import sklearn
from sklearn.datasets import fetch_openml
mnist = sklearn.datasets.fetch_openml("mnist_784", version=1, return_X_y=True)
pixel_data, labels = mnist
```

In [3]:
```
Index_3 = np.argwhere(labels == '3')[0:25,:]
Index_9 = np.argwhere(labels == '9')[0:25,:]
data_3 = pixel_data[Index_3.ravel(),:]
data_9 = pixel_data[Index_9.ravel(),:]
```

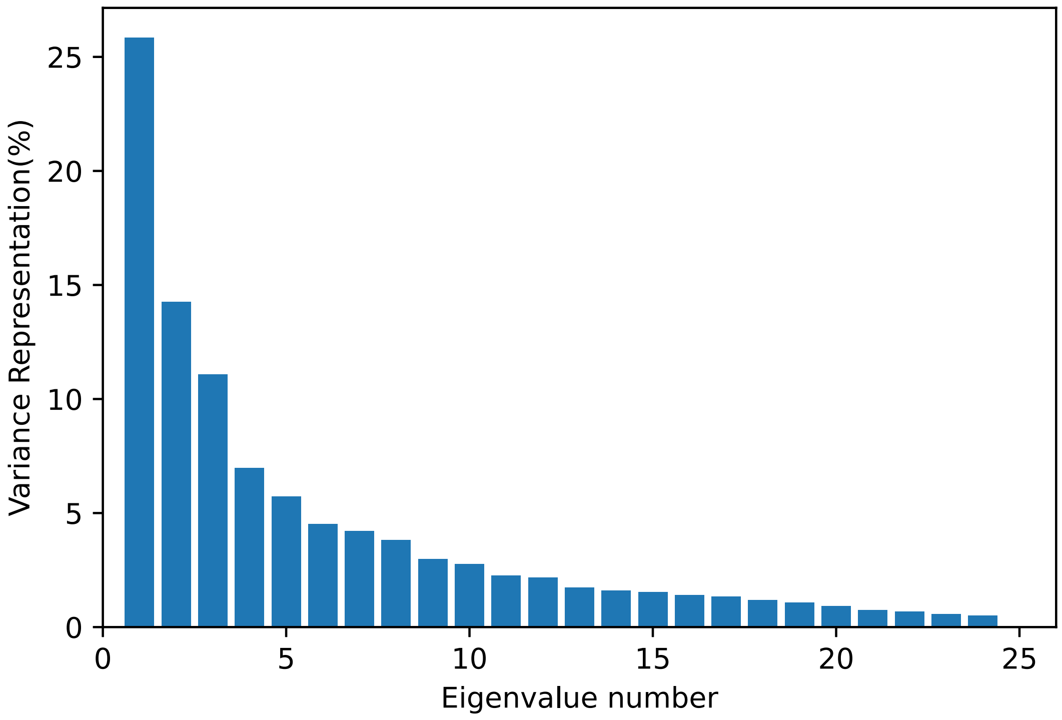
Digit 3

In [8]:
```
mean_3 = np.mean(data_3,axis = 0)
D_3 = data_3 - mean_3
n = len(data_3)
cov_3 = D_3@D_3.T
v,d = np.linalg.eig(cov_3)
idx = v.argsort()[::-1]
v = v[idx]
d = d[:,idx]
print("eigenvalues: ",v)
print("eigenvector: ",d)
# normalize eigenvalue
v_per = v/np.sum(v)
m = np.arange(1,26,1)
plt.bar(m,v_per*100)
plt.xlabel("Eigenvalue number")
plt.ylabel("Variance Representation(%)")
plt.xlim(0,26)
plt.show()
```

```
eigenvalues:  [ 1.75977789e+07  9.71031427e+06  7.53835004e+06  4.75560635e+06
  3.90048911e+06  3.07676079e+06  2.86770401e+06  2.59588136e+06
  2.03153580e+06  1.88330176e+06  1.53442185e+06  1.47475342e+06
  1.18297269e+06  1.08598316e+06  1.04735047e+06  9.64252580e+05
  9.10892037e+05  8.15182900e+05  7.39095866e+05  6.28443381e+05
  5.08898459e+05  4.66549439e+05  3.91610814e+05  3.51882671e+05
 -1.18145040e-09]
eigenvector:  [[-2.68247008e-01 -4.56606667e-02  8.22784105e-02 -5.54251034e-02
  -7.03867070e-02 -1.39259668e-01 -5.69725029e-02 -3.02323827e-02
  -6.01488936e-02 -1.04523199e-01  1.71108637e-01  1.08420902e-01
   2.10732276e-01 -1.34433959e-02 -1.55003736e-01 -2.91248616e-01
   8.13758062e-02 -4.93648935e-01  4.93168318e-02  2.94071840e-01
  -2.33956607e-01 -2.73241305e-02  4.90988336e-01  5.30946811e-02
  -2.00000000e-01]
 [ 1.34135476e-02 -2.43374713e-01 -1.86491419e-01  2.31926980e-01
   9.76372393e-02  1.16003517e-01  2.17433686e-01 -4.67667562e-02
   1.87381040e-01 -2.30931149e-01 -4.35802101e-02 -2.08317744e-02
   3.76498328e-01 -1.16947099e-01 -3.34126199e-02 -2.44937046e-01
   1.91493070e-01  4.16382708e-01  4.11910716e-01  1.49045162e-01
   8.61864647e-02 -1.50408665e-01 -6.54967582e-03 -1.51730358e-02
  -2.00000000e-01]
 [ 2.44633923e-01 -3.03293407e-02  3.12279619e-01  8.99148020e-02
   2.27388801e-01  6.24960256e-02  3.25677458e-01 -2.13508683e-01
   2.92883373e-02  2.49299646e-02 -5.30017424e-02  2.00974917e-01
  -5.83318769e-01  1.26894553e-01 -2.81907709e-01 -2.89923670e-01
  -6.21624341e-02  1.96439850e-02  8.96012708e-02  3.48578801e-03
   6.25478515e-03  9.64484880e-04  7.30442598e-02  6.42785565e-02
  -2.00000000e-01]
 [-1.87169645e-01 -4.40657973e-02  3.17535049e-01  1.64668008e-01
   1.52684165e-01  1.04227620e-02  1.12570603e-01 -2.46063487e-01
  -1.05515138e-01  2.69368648e-02  3.22543010e-02 -2.24183503e-01
   1.93514679e-01 -6.14478037e-02 -2.44925510e-01  5.86405732e-01
  -3.65790011e-01 -5.59954203e-02  1.86527635e-01  3.53936477e-02
   9.86240523e-02 -4.33382306e-03  3.67725144e-02 -7.82229854e-02
  -2.00000000e-01]
 [ 2.18241077e-01  1.62603539e-01 -1.15216847e-01 -3.47692381e-02
   5.15595616e-01  3.91175696e-01 -4.70795749e-01 -9.02222783e-02
  -2.73882368e-01 -2.74052088e-01 -1.08684071e-01  1.17456870e-01
   9.43685126e-02  8.63461930e-02  1.14578628e-01 -6.77727295e-05
  -7.33383806e-02 -6.11992462e-02 -2.78082344e-02  6.03007933e-02
  -2.32657494e-02  3.62704050e-02  1.43978338e-02  4.46078961e-02
  -2.00000000e-01]
 [ 2.03242048e-01  1.22383562e-01 -1.23678485e-01  4.52736899e-02
```

```
 -2.74760520e-01 -1.85231919e-02 -4.25601317e-01  3.80779840e-02
 -8.57117898e-02  3.00006945e-01  9.83785397e-02 -2.43489415e-01
  1.27068580e-02  1.37043213e-01 -4.91180042e-01 -8.91649168e-02
 -1.55796735e-02  3.09068425e-01  3.60129152e-02 -1.37319898e-01
  5.64001286e-02 -8.25419877e-02  2.42378124e-01  9.39060643e-02
 -2.00000000e-01]
[-1.07305263e-01 -1.97321865e-01 -1.43510079e-02  2.03776330e-01
  1.69958548e-02 -1.67146556e-01 -5.16919488e-02 -3.96124105e-01
 -1.34777091e-01  9.09664651e-02  6.70931220e-02  3.02951445e-01
  1.95851892e-01 -2.35269124e-01 -9.14275321e-02 -1.24560335e-01
  5.59804111e-02  8.85161698e-02 -5.83728841e-01 -1.89402837e-01
  7.24345906e-02 -5.59103853e-02 -2.08125029e-01  2.07513713e-02
 -2.00000000e-01]
[ 1.66632457e-01 -1.81048811e-01 -1.23450443e-01  1.53092715e-01
  4.89072697e-02 -1.80402980e-01  2.13591154e-01  4.69791781e-01
 -3.78877201e-01 -1.51761988e-02 -4.14344719e-01  1.00294226e-01
  8.75885251e-05  2.26880367e-03 -1.50199986e-01  2.86945298e-01
  2.97049998e-01 -9.32360069e-02 -8.74397728e-02 -7.88656732e-02
 -7.30316786e-02 -4.07458073e-03  1.13778178e-01 -7.20207612e-04
 -2.00000000e-01]
[ 1.61252691e-01 -7.95280577e-02 -1.16474922e-03  3.01547735e-01
 -9.57657942e-02 -1.81332815e-01 -7.07828230e-02  1.07161585e-02
  3.46061882e-01 -2.31304391e-01  1.67498623e-01  3.12325491e-02
  3.21980974e-02  4.19254913e-01  2.01562886e-01  1.84212046e-01
  1.66235519e-02 -1.96479441e-01 -1.99861548e-02 -1.68817200e-01
  1.44159928e-01 -2.74611518e-04 -2.30210582e-02  4.96796261e-01
 -2.00000000e-01]
[ 3.14095009e-01  1.38980042e-01 -1.11673677e-01 -3.01149771e-01
  4.54015580e-01 -4.71577309e-01  1.21220335e-01 -5.19022194e-02
  1.53528194e-01  1.22637111e-01  2.39615792e-01 -3.53673246e-01
  5.33603716e-02 -1.45652770e-01  7.32943575e-02  1.91605665e-02
  1.61356327e-01 -3.53819589e-02 -7.91507850e-02  3.49358707e-02
 -2.67648969e-02 -3.50013208e-02 -5.00808635e-03 -7.28766955e-02
 -2.00000000e-01]
[ 1.66529466e-01 -8.66390890e-02  5.38177097e-02  2.60157647e-01
 -8.82065207e-02 -1.35461686e-01 -2.11803250e-01  2.46958172e-01
  2.73318711e-01 -1.13432903e-01  3.24741871e-02  6.52249543e-02
 -4.53646468e-02  1.20456332e-01 -2.20954745e-02 -4.34560039e-02
 -2.69478771e-01 -3.74456070e-02 -1.22126718e-01  2.08239543e-01
 -1.93508361e-01 -3.79495756e-02 -1.96312124e-01 -6.32974717e-01
 -2.00000000e-01]
[-1.98324727e-01 -2.39643236e-01 -4.09803302e-02 -2.86402507e-01
  8.35585877e-03  2.67410113e-02 -2.90240890e-02  2.38719704e-01
 -5.15132151e-02  1.57138710e-02 -1.08251016e-01 -2.01631826e-01
 -8.99301546e-02  7.30635283e-02 -1.38065690e-01 -1.48664382e-01
 -1.76282703e-01 -7.63456635e-02 -1.58488037e-01  4.29781592e-01
  3.94566825e-01 -1.10611011e-01 -3.63882275e-01  2.52966544e-01
 -2.00000000e-01]
[-1.63782636e-01  4.07592545e-01  4.17802914e-01 -1.01883261e-01
 -8.48539231e-02 -5.22093886e-03 -4.11332585e-02 -2.56190459e-02
  1.38971502e-01 -2.92952384e-02 -1.48453962e-01  9.78208666e-02
  8.11271457e-02  1.89148006e-01  1.53315055e-02  1.53022752e-01
  4.59481891e-01  3.07548166e-01 -1.58978189e-01  3.05633198e-01
  2.54288125e-03  1.66199811e-01 -4.94468825e-02 -1.64602327e-02
 -2.00000000e-01]
[ 8.52409263e-02 -7.13928386e-02 -1.20203211e-01  2.05863657e-01
 -2.12209445e-01 -6.05764969e-03 -1.39173738e-01 -2.81804623e-01
 -2.32978280e-01  3.24426804e-01  3.46632753e-02 -3.70892287e-02
 -2.05291743e-01 -5.00049995e-02  3.30393547e-01  3.41075144e-02
  2.90461516e-01 -2.38954138e-01  3.04015547e-01  2.14326324e-01
  2.85838934e-01  1.43370204e-01 -9.78569477e-02 -1.95487538e-01
 -2.00000000e-01]
[-2.07780498e-01 -2.27868305e-01 -1.51025724e-01 -2.31120413e-01
 -1.57968762e-02  9.76027751e-02 -1.98260223e-01 -1.58324775e-01
  2.88775613e-01  3.58084662e-02 -1.54914515e-01  6.61478069e-02
 -3.43910765e-01 -1.10217073e-01  1.00508914e-01  3.13995201e-01
  1.29466494e-01  6.33471076e-02  8.18651637e-03 -3.09519676e-02
 -2.48719574e-01 -5.10888807e-01  1.32757315e-01  8.38627898e-03
 -2.00000000e-01]
[ 2.37778161e-01  1.25245640e-02  1.10519040e-01 -1.81856083e-02
 -1.51316938e-01  2.14964427e-01 -6.59182217e-02  2.56922602e-01
  2.18639948e-01  2.94770544e-03  1.71889895e-01  1.61392442e-01
 -6.08975741e-02 -6.88559461e-01 -1.91323966e-02  1.29718096e-01
 -6.19811138e-02 -1.87379692e-02  2.21794273e-03  7.90765250e-02
 -7.12072218e-03  2.90423419e-01  1.71758544e-02  2.40281162e-01
 -2.00000000e-01]
[ 4.94493454e-02 -9.08537930e-02  4.57713839e-01 -2.00156280e-01
  1.27539664e-03 -4.42527281e-02 -1.41823046e-01  1.10735044e-01
  5.14582495e-02  2.97088753e-01 -3.16977683e-01  2.81147911e-02
  2.83878988e-01 -1.42677343e-02  2.03208853e-01 -2.53515557e-01
 -6.91332385e-02 -1.29992501e-01  2.58965071e-01 -3.92039439e-01
 -6.47590937e-02 -1.05656949e-01 -1.68716386e-01  2.47481023e-02
 -2.00000000e-01]
[-2.47506775e-01 -1.97497901e-01 -1.25914206e-01 -3.33356104e-01
  8.23789784e-03  3.72873771e-02 -6.18433286e-02  1.30173803e-03
  9.43934731e-02 -2.78388352e-01  1.29813663e-01  2.60427913e-02
 -9.89571552e-02  9.59443506e-02 -2.38538533e-01  2.44591501e-02
  1.81638281e-01 -5.72134927e-02  1.24824808e-01 -4.22757549e-01
  1.59143188e-01  4.61875231e-01 -7.77730869e-02 -2.67877013e-01
 -2.00000000e-01]
```

```
[-2.13320877e-01  1.98014254e-01  1.43071525e-01  2.33202414e-02
  2.66622342e-02 -2.11736322e-02  6.19694540e-02  3.44353952e-01
 -3.35071715e-01 -9.71210717e-02  5.23394217e-01  1.60296080e-01
 -1.36846263e-01 -2.53557698e-02  2.18119124e-01 -1.91699279e-02
 -1.04884479e-02  1.73983431e-01  8.74490721e-02 -1.73529782e-01
  1.72770528e-01 -3.70877862e-01  3.48252635e-03 -8.05277584e-02
 -2.00000000e-01]
[-2.37159263e-01  3.82614122e-01 -1.74069366e-01  1.92551136e-01
  2.80288950e-02 -2.41689838e-04  8.93374318e-02  4.55422532e-02
  2.27144315e-01 -2.91572028e-02 -3.79419363e-01 -1.54388643e-01
 -8.49756520e-02 -1.66328451e-01  1.55322064e-01 -1.62624962e-01
 -1.87935886e-01 -7.54196113e-02 -1.66648129e-01 -1.75570136e-01
  3.81538870e-01  1.48857564e-02  3.37653889e-01 -1.18487668e-01
 -2.00000000e-01]
[ 2.28623149e-01 -3.73025394e-02  9.80450274e-02 -7.90006754e-02
 -3.02226932e-01  4.95513350e-01  3.07464946e-01 -9.13347547e-02
 -8.78634573e-02 -1.51323811e-01  9.38215097e-02 -4.38993261e-01
  8.81053858e-02  6.49551530e-02  9.52807615e-02 -4.96388080e-02
  1.23095586e-01 -1.53424571e-01 -3.08487729e-01 -1.14858023e-01
 -1.23199459e-01 -1.36883149e-01 -6.02660122e-04 -1.18785410e-01
 -2.00000000e-01]
[-2.26377997e-01  2.47983667e-01 -1.15290004e-01  1.71488925e-01
 -1.16076251e-01 -2.00937560e-01 -2.60419666e-02 -8.30454365e-02
 -1.94235461e-01 -2.36711085e-01 -1.09288354e-01 -2.84743555e-01
 -1.99579058e-01 -1.47181074e-01  4.56352809e-03 -1.27501725e-01
 -6.98159156e-02  3.69445048e-02  1.55908588e-01 -6.11229787e-02
 -4.92527817e-01  1.15276060e-01 -3.73490377e-01  2.21198146e-01
 -2.00000000e-01]
[-1.53655276e-01  2.64551467e-01 -3.62131573e-01  4.18050479e-02
  1.21262004e-01  3.00567293e-01  2.66170464e-01  1.02945677e-01
  1.71034286e-01  4.84619007e-01  1.31874381e-01  2.37942190e-01
  1.47958404e-01  2.01978145e-01 -1.31222857e-01  5.68320846e-02
 -3.25358047e-02 -1.67452195e-01  3.85190358e-02 -1.45965225e-02
 -1.99760571e-01  2.70880925e-02 -2.30875271e-01  3.22036644e-02
 -2.00000000e-01]
[-1.38351057e-01 -3.36821357e-01 -1.63015082e-02 -2.55858958e-02
  7.94560466e-02  1.59384835e-02  6.77540916e-02  6.84115774e-02
 -9.79372131e-02  2.32369488e-01  6.44790097e-02 -9.17945178e-02
 -4.74320427e-02  1.82483611e-01  4.01969272e-01 -3.53578142e-02
 -2.44128039e-01  3.76241291e-01 -1.43408947e-01  7.02147747e-02
 -2.24398910e-01  4.01561447e-01  3.05014572e-01  6.93449793e-02
 -2.00000000e-01]
[ 2.59849222e-01  1.72100550e-01 -2.11120582e-01 -4.18352059e-01
 -3.74902953e-01 -1.97124313e-01  2.07675840e-01 -2.19528096e-01
 -1.41483730e-01 -1.67034755e-01 -1.21443519e-01  3.46506138e-01
  1.26115294e-01  7.48379529e-02  8.29786483e-02  9.09730945e-02
 -3.49372515e-01  9.92509683e-02  1.02795587e-01  7.53269472e-02
  5.05522637e-02 -2.51780510e-02  3.42164576e-02 -2.49704457e-02
 -2.00000000e-01]]
```



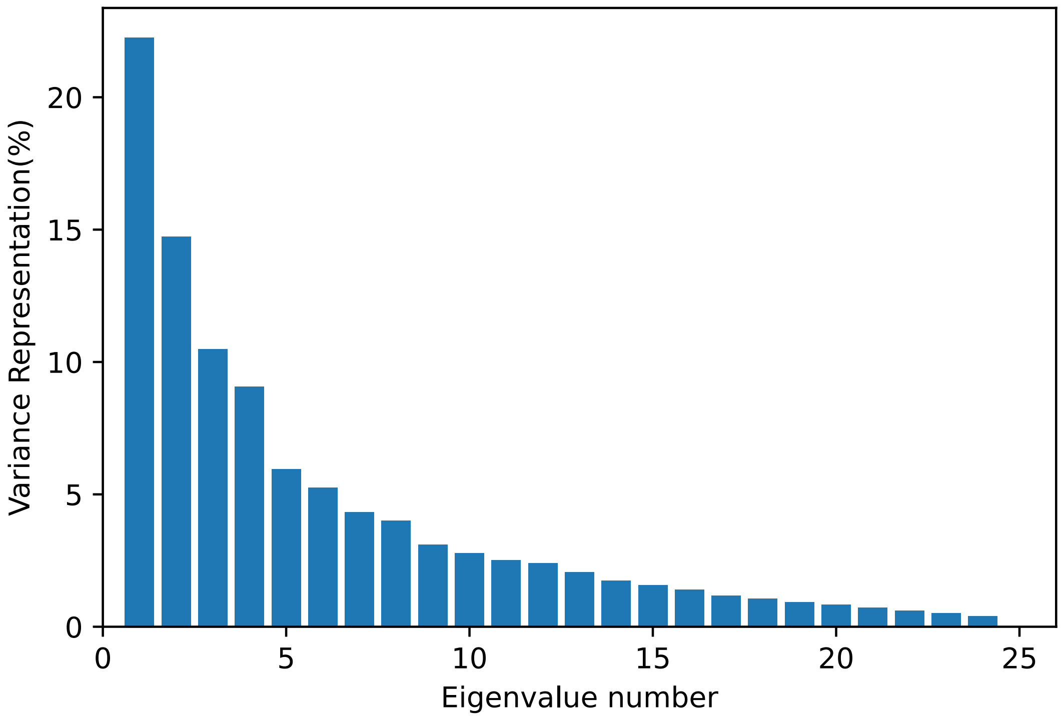
Digit 9

In [9]:
```
mean_9 = np.mean(data_9,axis = 0)
D_9 = data_9 - mean_9
n = len(data_9)
cov_9 = D_9@D_9.T
v,d = np.linalg.eig(cov_9)
idx = v.argsort()[::-1]
v = v[idx]
d = d[:,idx]
print("eigenvalues: ",v)
print("eigenvector: ",d)
# normalize eigenvalue
v_per = v/np.sum(v)
m = np.arange(1,26,1)
plt.bar(m,v_per*100)
plt.xlabel("Eigenvalue number")
plt.ylabel("Variance Representation(%)")
plt.xlim(0,26)
plt.show()
```

```
eigenvalues:  [1.11489437e+07 7.38233800e+06 5.25487751e+06 4.54628343e+06
 2.98103687e+06 2.63526606e+06 2.16869120e+06 2.00765892e+06
 1.56007603e+06 1.39097015e+06 1.25967396e+06 1.20374980e+06
 1.03335590e+06 8.76043667e+05 7.85822811e+05 7.02084048e+05
 5.93920641e+05 5.33846815e+05 4.67261731e+05 4.21275998e+05
 3.67451360e+05 3.05117489e+05 2.55962413e+05 2.04917246e+05
 1.28804569e-09]
eigenvector:  [[-3.99557808e-02 -2.02151669e-01  1.46429503e-01  4.12717072e-02
  -2.85635693e-01  1.63062696e-01 -1.88741402e-01 -3.47968385e-01
   9.92564979e-02 -6.96518457e-02  1.81942918e-01  4.89392021e-01
   2.56689061e-01 -5.86224983e-02  2.43794556e-01  1.81857592e-01
   2.10956191e-01  1.04577767e-01 -7.76093905e-02  1.77327823e-01
  -2.13250769e-03 -1.90217132e-01 -1.07032697e-01 -1.99538041e-01
  -2.00000000e-01]
 [ 3.02049323e-01  6.59504977e-02  7.31793517e-02  1.17247462e-01
   9.36569281e-02  8.89984277e-02  8.39855180e-02 -1.21042470e-01
   1.33471041e-02 -2.74898636e-01 -1.38373710e-01 -1.03761783e-01
   3.61060351e-01  6.03994119e-02  1.79808279e-01 -4.00428141e-01
  -4.60648002e-02  2.86088342e-02 -3.02342861e-01  2.45432045e-01
   1.25573244e-01  1.50650165e-01 -4.13219391e-02  4.24039680e-01
  -2.00000000e-01]
 [ 2.56915263e-01  8.35431614e-02  8.31690182e-02 -4.51716617e-03
   8.32458436e-02  6.07118117e-02  3.53724804e-01 -3.29233867e-01
  -4.19612751e-01 -1.31773714e-02 -2.12638939e-01  1.89670280e-01
  -3.20876973e-01 -1.97027699e-01 -2.47343205e-01  1.26005635e-01
   1.51323360e-01 -8.71298041e-02  1.25864576e-01  2.47633941e-01
  -1.06029920e-01  1.90492603e-01 -2.57565765e-02 -6.18139324e-02
  -2.00000000e-01]
 [ 2.71987372e-01  1.52309471e-01 -5.22764790e-02 -2.64350715e-02
   1.76360102e-01  3.73432978e-01 -7.98295941e-02 -6.77530851e-02
   1.78116502e-01 -3.85974898e-01  1.90855109e-01 -2.79283397e-01
  -1.39946014e-02  2.96937109e-01  4.88487221e-02  2.47867622e-01
   6.03229376e-02  8.00120973e-02  2.70243213e-01 -1.56453807e-01
  -3.14976422e-01  5.63790058e-02  5.63386689e-02 -1.32071639e-01
  -2.00000000e-01]
 [ 2.55144934e-01  1.00550411e-01  9.29025227e-02  1.89399629e-01
  -1.43126044e-01  1.15947072e-01  2.16284533e-02  2.07084884e-01
   6.61372569e-03  3.01742863e-01 -3.14823922e-02  1.32324975e-01
  -2.85097666e-01 -5.00647626e-02  1.85981322e-01 -2.49807899e-01
   2.31993893e-01  1.01781166e-01 -3.09735415e-01 -4.16750466e-01
  -3.48278460e-01 -1.11445669e-01  5.81060960e-02  4.17280059e-02
  -2.00000000e-01]
 [-4.33503075e-02 -2.62800681e-01 -5.40909450e-02 -1.30610500e-01
  -2.99973497e-01 -8.85948765e-03 -2.54117344e-01  1.81951378e-01
  -4.46085690e-02 -1.59807687e-01 -6.70409007e-02  1.22693649e-01
  -1.89287443e-01 -1.73557330e-01  1.19383302e-01  1.61454163e-01
  -1.22662451e-01  1.38979134e-01  1.81399606e-01 -2.09906867e-02
  -1.52659105e-02  3.91070441e-01  3.75161225e-01  4.07867601e-01
  -2.00000000e-01]
 [-9.45728323e-02  3.93382246e-01  2.49941514e-01  3.89647927e-02
   2.37936489e-01 -3.29802150e-01 -1.40345502e-01  3.47405361e-01
  -4.55812874e-01 -1.42509609e-01  2.02286760e-01  2.66679914e-01
   1.55331335e-01  1.47682595e-01  1.41917284e-01  1.02872556e-01
  -7.30458451e-03  4.56047764e-02  7.51758368e-02 -2.79151098e-02
   1.39239867e-02 -7.23749644e-03 -3.10314953e-02 -2.16284906e-04
  -2.00000000e-01]
 [-1.11425618e-01  1.04789694e-01 -4.83341908e-01  7.74562271e-02
  -1.57229722e-01 -1.62681370e-01  3.60168308e-01  1.67953183e-01
   1.71602831e-01 -2.63330316e-01  2.25251379e-01  1.58643610e-01
   1.56403637e-01 -1.99600868e-01 -1.85574422e-01  1.38275259e-01
  -2.81423633e-03 -3.01378827e-01 -1.92071656e-01 -4.76424370e-02
  -2.26864102e-01 -2.30084550e-02 -6.27506243e-02  9.68453153e-02
  -2.00000000e-01]
 [ 1.94117298e-01 -1.74086141e-01  2.44618878e-02 -5.67875978e-02
   1.02820994e-01 -1.88333879e-01  2.93095917e-02 -1.03747112e-01
   7.97943498e-04  8.49864514e-02 -2.69982929e-01 -8.12613544e-02
  -3.18604195e-02  2.53162004e-02  1.79868671e-01  4.18956648e-01
  -2.19979488e-01 -1.38868319e-01  6.76294314e-02 -1.99220944e-01
   4.67887208e-02 -4.77043673e-01 -2.55084064e-01  3.57082879e-01
```

```
 -2.00000000e-01]
[-5.49044422e-02  5.32503165e-01  2.32534433e-01 -5.31863138e-01
 -2.56853294e-01 -2.34775457e-01 -6.60251179e-02 -2.17190601e-01
  3.42135550e-01  2.86034387e-03 -7.49542827e-02 -9.43080708e-02
 -8.59071875e-02 -1.06171836e-01 -3.82838520e-02 -8.75923368e-02
  1.66699352e-02 -4.08475103e-02 -1.01403815e-02 -2.07226202e-02
  4.04198848e-02 -7.41268885e-03  1.62952139e-02 -7.10370266e-03
 -2.00000000e-01]
[ 1.70149974e-01 -2.13189462e-01 -9.09715303e-02 -3.02386824e-01
  2.38037832e-01 -2.71234775e-02 -4.69751500e-03 -4.27217604e-02
  8.64613934e-02  3.43533373e-02  2.88720839e-01  3.01000037e-01
 -1.91015723e-01  3.17581624e-01 -4.07933968e-01 -7.97465456e-02
 -1.74771273e-01  2.26388429e-01 -2.68948714e-01 -1.10848360e-01
  2.57448114e-01 -9.61466061e-03  1.78029591e-02  2.83393034e-02
 -2.00000000e-01]
[-1.94595507e-01 -1.28959320e-01  2.95934622e-01 -3.02201195e-01
 -1.28071418e-01  4.00250598e-01  3.38847818e-01  4.76448108e-01
 -7.18634715e-03  1.23847947e-01 -1.84155232e-01 -7.34580507e-02
  2.02982424e-01  1.75842910e-01 -1.32431340e-01  1.41836893e-01
  1.17502935e-01 -1.59072763e-02 -4.89615809e-02  1.50323718e-01
  1.64708759e-02 -4.61320769e-02 -4.72455810e-02  3.48658187e-03
 -2.00000000e-01]
[-2.03281147e-01 -2.32919721e-01  1.03686707e-01 -1.45113515e-01
  1.96834070e-01  5.75958849e-02 -1.29122149e-01 -1.74741551e-01
 -2.55736590e-01 -1.21230346e-01  9.44918385e-02 -1.82903090e-01
  2.10456920e-01 -2.92509609e-01 -2.27709134e-01 -2.70094913e-01
  2.04210607e-01 -1.94514141e-01  6.32705178e-02 -2.81073521e-01
 -5.69227260e-02 -2.97580129e-01  3.31224821e-01  4.65196056e-02
 -2.00000000e-01]
[ 1.19818099e-01  3.98158886e-02 -8.15978668e-02  2.11727363e-01
 -2.64261088e-01  5.19411868e-03 -3.33972684e-01  2.02432021e-01
 -1.33044333e-01 -3.15752082e-01 -2.89825172e-01 -2.19378652e-01
 -8.96529755e-02 -1.46467065e-01 -3.02093451e-01  5.98297073e-02
 -5.23121947e-02  1.50877869e-01 -2.24738017e-01  1.42111419e-02
  2.55989549e-01 -1.61781967e-01 -1.35107292e-01 -3.31919099e-01
 -2.00000000e-01]
[ 2.08777222e-01  1.90856865e-02  1.30882462e-01  1.82201582e-01
 -5.80626714e-02  3.00985388e-02  1.57876595e-01 -2.14520279e-02
  9.24568134e-02  1.90478550e-01 -7.47758629e-02  1.12245430e-01
  2.73614762e-01 -3.44130774e-02  4.05139101e-02 -2.47008449e-03
 -5.02943251e-01 -2.53121705e-01  1.16012989e-01 -2.17857484e-01
  1.74449357e-01  1.48551766e-01  3.55294565e-01 -3.72373930e-01
 -2.00000000e-01]
[-2.99905118e-01  2.38663438e-01 -3.64296210e-01  2.70490147e-02
  8.66927581e-02  1.93119132e-01  1.77098325e-01 -7.99460838e-02
 -5.22988615e-02  9.56610888e-02 -1.52865778e-01  7.21649230e-02
  5.32213201e-03 -7.92864238e-02  1.10268496e-01 -1.55574830e-01
 -2.57323393e-01  5.36249364e-01  1.98211629e-01  1.04526078e-01
 -2.16596147e-02 -3.03792626e-01  1.30873254e-01  8.30932200e-03
 -2.00000000e-01]
[-2.45789205e-01 -8.38844735e-02  1.27644353e-01  1.41394430e-01
 -1.03418208e-02 -1.74974725e-01  1.99478792e-01 -1.90884965e-01
 -1.00607603e-01  7.54142589e-02  2.73016545e-01 -4.46407582e-01
 -1.83352619e-01  6.84918813e-02  2.10967953e-01  2.70260449e-01
 -5.52533768e-02  1.16070753e-01 -4.35318489e-01  1.48922933e-01
  1.31373225e-02  5.94963409e-02  2.52934378e-01 -8.90807103e-02
 -2.00000000e-01]
[ 1.21981543e-01 -2.04805811e-01 -1.83929183e-01 -2.07933153e-01
  3.31992004e-01 -3.04024308e-01 -9.80052088e-02  8.55168378e-02
  1.14641951e-01  2.35178870e-01 -1.93627329e-01 -1.04681169e-01
  3.12188886e-01 -2.37167936e-01  6.81429789e-02  7.63082333e-02
  1.70076545e-01  2.61923051e-01 -4.76920721e-02  2.90859493e-02
 -2.18860241e-01  3.01651927e-01 -1.34976002e-01 -2.60192925e-01
 -2.00000000e-01]
[-2.75625211e-01 -1.34102309e-01  1.88415282e-01 -8.31843898e-03
  9.75143350e-02  1.94005003e-01 -3.05619012e-02 -3.94817449e-03
 -7.34906721e-04 -8.55267284e-02  1.77483607e-01 -4.58667221e-02
 -2.13966497e-01 -2.56247512e-01  1.14322847e-01 -2.24399946e-01
 -3.89835751e-01 -8.89944829e-02  3.78632686e-02 -1.05571363e-01
 -1.00086677e-01  2.17027209e-01 -5.76638805e-01 -6.70135290e-02
 -2.00000000e-01]
[-2.31839041e-01  1.41846862e-01 -4.30952828e-01 -1.06132598e-01
  1.79219368e-02  2.29683115e-01 -2.72779233e-01 -1.56845223e-01
 -2.23308774e-01  2.17041763e-01 -1.91027585e-01  1.33401803e-02
 -1.03815131e-02  3.11181609e-01  1.64214084e-01  3.04604037e-02
  1.25760461e-01 -3.84229622e-01 -1.58607800e-01 -1.05957366e-01
  1.64581675e-01  2.17344481e-01 -6.26441510e-02 -2.23686365e-02
 -2.00000000e-01]
[-1.79925712e-01  7.76884073e-02  1.62606833e-01  3.71352961e-01
  1.53179660e-01 -6.29481533e-03 -3.50924622e-01 -3.06258669e-02
  2.67199018e-01  2.60080837e-01 -8.62340087e-02  4.52280789e-02
  1.89817343e-03  1.11098170e-01 -4.10184346e-01  5.32988390e-02
 -1.04400297e-01 -1.20756965e-01  8.20695234e-03  3.18797278e-01
 -3.27591424e-01 -1.61085418e-02  6.58302230e-02  1.96494272e-01
 -2.00000000e-01]
[-2.11795950e-03 -2.93436614e-01 -7.62574203e-02 -6.79420556e-02
 -6.63796461e-02 -3.16412303e-01  4.80163848e-02  1.15321149e-01
  5.44181059e-02 -1.82328865e-01 -1.35912810e-01  2.01967499e-02
 -2.46629033e-01  3.49818339e-01  2.11684222e-01 -3.83398163e-01
  8.12232812e-03 -2.24640726e-01  1.88727382e-01  2.97881932e-01
```

```
 -1.74453012e-01 -1.87260016e-01  8.11864656e-02 -2.57430849e-01
 -2.00000000e-01]
[ 2.51194747e-01  7.89356616e-02 -1.30354160e-01  1.65675778e-02
  1.41557986e-01  1.41609976e-01 -9.30225631e-02  2.52952117e-01
  1.29410881e-01  2.53213546e-01  3.76975477e-01 -1.18754392e-01
 -1.62747030e-01 -3.18085904e-01  1.43672232e-01 -1.46600880e-02
  1.65757884e-01 -1.80977122e-01  1.88224918e-01  3.38304180e-01
  3.78025143e-01 -1.41213279e-01  3.00449719e-02  3.49393715e-02
 -2.00000000e-01]
[-2.69565865e-01 -3.78690085e-02  1.10013170e-01  3.47899893e-01
  1.73302751e-01 -1.20658835e-01  2.10063329e-01 -3.21675466e-02
  3.42474667e-01 -1.43037197e-01 -1.76589411e-01  7.43228094e-02
 -1.18914627e-01  6.00951818e-02 -2.85827974e-03  1.99610556e-03
  3.96524298e-01  8.35476698e-02  2.00327037e-01 -2.73512257e-01
  4.05295245e-01  1.55969676e-01 -4.12957632e-02  3.66191677e-02
 -2.00000000e-01]
[ 9.47179713e-02 -6.08593802e-02 -7.37331288e-02  1.27708614e-01
 -4.61118796e-01 -1.79768544e-01  6.19469156e-02 -1.16796321e-01
 -2.05981374e-01  2.82365726e-01  2.68461869e-01 -2.47838394e-01
  2.07736627e-01  2.24777489e-01 -2.08976861e-01 -1.43107159e-01
  7.64437212e-02  1.56745590e-01  3.55009020e-01 -8.79305986e-02
  2.10178986e-02  9.12147943e-02 -2.50207852e-01  1.18852174e-01
 -2.00000000e-01]]
```

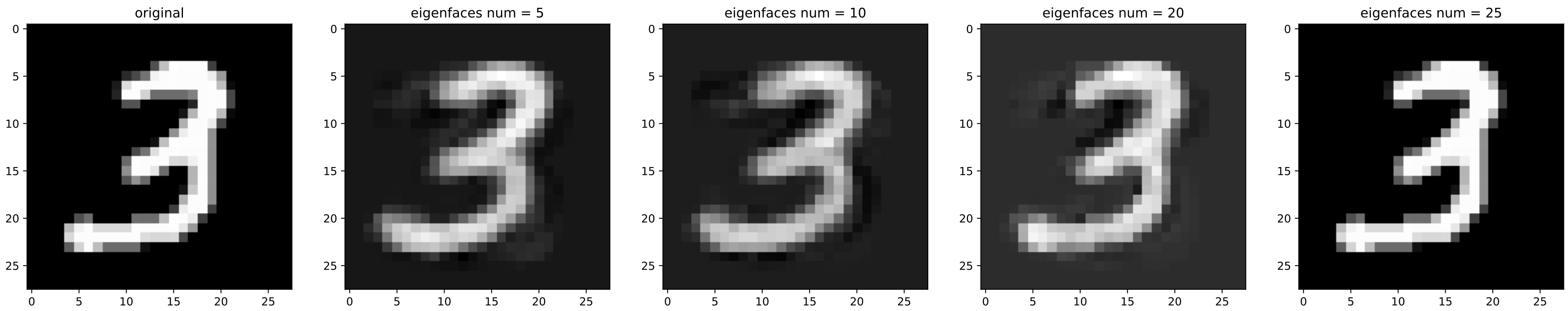


(b) Reconstructiong Training Digit

digit 3

In [ ]:
```
from sklearn.decomposition import PCA
fig,axs = plt.subplots(1,5,figsize = (25,5))
pic = data_3[11,:].reshape(28,28)
axs[0].imshow(pic,cmap = "gray")
axs[0].set_title("original")
eigen_num = [5,10,20,25]
for i in eigen_num:
    pca = PCA(n_components = i)
    pca.fit(data_3)
    A =  pca.transform(data_3)
    approx = pca.inverse_transform(A)
    mse = np.mean((approx[11,:]-data_3[11,:])**2)
    print(f"mse(using {i} eigenvectors): ",mse)
    pic = approx[11,:].reshape(28,28)
    axs[eigen_num.index(i)+1].imshow(pic,cmap = "gray")
    axs[eigen_num.index(i)+1].set_title(f"eigenfaces num = {i}")
```

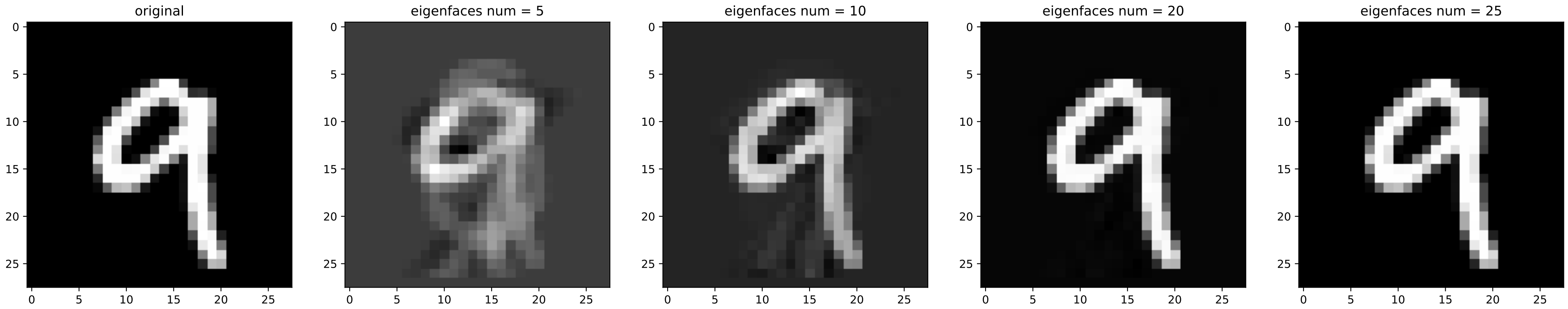
```
mse(using 5 eigenvectors):  790.5380029279061
mse(using 10 eigenvectors):  588.8437987667844
mse(using 20 eigenvectors):  203.19639798612326
mse(using 25 eigenvectors):  7.669059676529222e-27
```



digit 9

In [ ]:
```
fig,axs = plt.subplots(1,5,figsize = (25,5))
pic = data_9[11,:].reshape(28,28)
axs[0].imshow(pic,cmap = "gray")
axs[0].set_title("original")
eigen_num = [5,10,20,25]
for i in eigen_num:
    pca = PCA(n_components = i)
    pca.fit(data_9)
    A =  pca.transform(data_9)
    approx = pca.inverse_transform(A)
    mse = np.mean((approx[11,:]-data_9[11,:])**2)
    print(f"mse(using {i} eigenvectors): ",mse)
    pic = approx[11,:].reshape(28,28)
    axs[eigen_num.index(i)+1].imshow(pic,cmap = "gray")
    axs[eigen_num.index(i)+1].set_title(f"eigenfaces num = {i}")
```

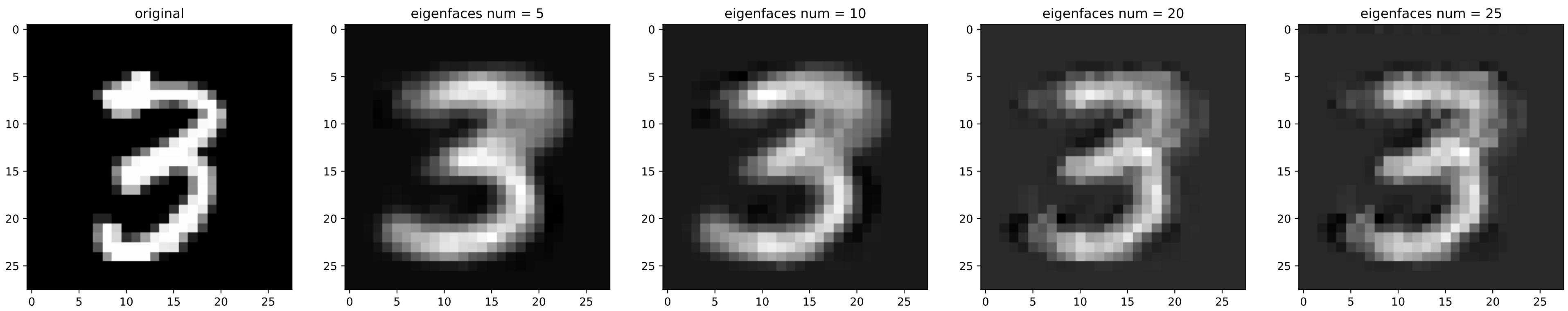
```
mse(using 5 eigenvectors):  1677.7787689670413
mse(using 10 eigenvectors):  213.01783392441948
mse(using 20 eigenvectors):  1.6873258300584355
mse(using 25 eigenvectors):  4.716830583600343e-27
```



(c) Reconstructing Test Digit

In [ ]:
```
fig,axs = plt.subplots(1,5,figsize = (25,5))
index = np.argwhere(labels == '3')[224,:]
test = pixel_data[index.ravel(),:]
pic = test.reshape(28,28)
axs[0].imshow(pic,cmap = "gray")
axs[0].set_title("original")
eigen_num = [5,10,20,25]
for i in eigen_num:
    pca = PCA(n_components = i)
    pca.fit(data_3)
    A =  pca.transform(test)
    approx = pca.inverse_transform(A)
    mse = np.mean((approx-test)**2)
    print(f"mse(using {i} eigenvectors): ",mse)
    pic = approx.reshape(28,28)
    axs[eigen_num.index(i)+1].imshow(pic,cmap = "gray")
    axs[eigen_num.index(i)+1].set_title(f"eigenfaces num = {i}")
```

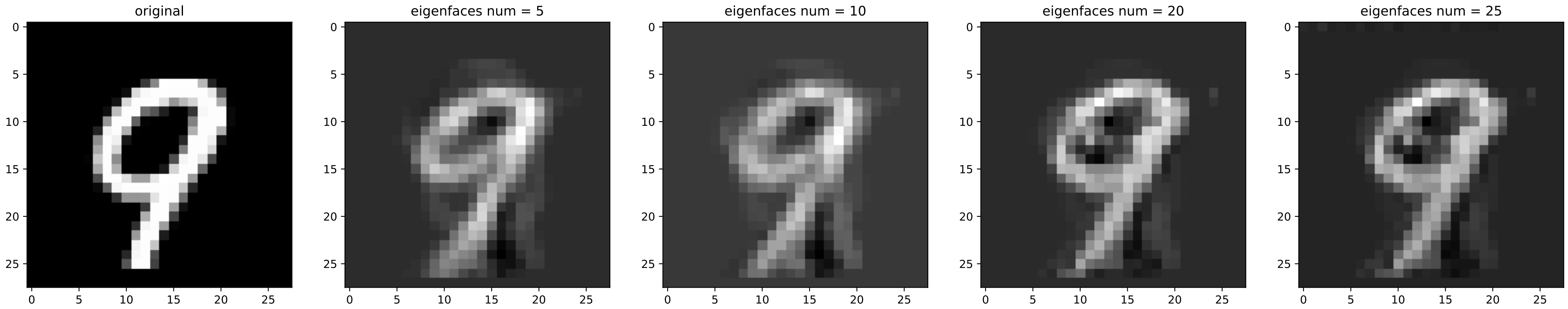
```
mse(using 5 eigenvectors):  2388.9571269534276
mse(using 10 eigenvectors):  2185.700230193912
mse(using 20 eigenvectors):  1621.7823254798086
mse(using 25 eigenvectors):  1596.2930327695233
```



digit 9

In [ ]:
```
fig,axs = plt.subplots(1,5,figsize = (25,5))
index = np.argwhere(labels == '9')[224,:]
test = pixel_data[index.ravel(),:]
pic = test.reshape(28,28)
axs[0].imshow(pic,cmap = "gray")
axs[0].set_title("original")
eigen_num = [5,10,20,25]
for i in eigen_num:
    pca = PCA(n_components = i)
    pca.fit(data_9)
    A =  pca.transform(test)
    approx = pca.inverse_transform(A)
    mse = np.mean((approx-test)**2)
    print(f"mse(using {i} eigenvectors): ",mse)
    pic = approx.reshape(28,28)
    axs[eigen_num.index(i)+1].imshow(pic,cmap = "gray")
    axs[eigen_num.index(i)+1].set_title(f"eigenfaces num = {i}")
```

```
mse(using 5 eigenvectors):  2500.6620267556564
mse(using 10 eigenvectors):  2283.668075902942
mse(using 20 eigenvectors):  1402.6012760585754
mse(using 25 eigenvectors):  1375.7686795727036
```



(d) Discussion

For training set, it will be lossless if using all eigenvectors to reconstruct data point because all the information is stored in the model. However, model doesn't have information of test data point, when reconstruct test point, there will be error.

Q2: PCA, Kernel PCA and t-SNE

(a) KNN

In [3]:
```
labels = np.array(list(map(int,labels)))
X_train, X_test,y_train,y_test = train_test_split(pixel_data, labels,test_size=0.4, random_state=42)
scaler = StandardScaler()
scaler.fit(X_train)
X_train_norm = scaler.transform(X_train)
X_test_norm = scaler.transform(X_test)
```

In [8]:
```
from sklearn.neighbors import KNeighborsClassifier
cluster_num = [2,10,20]
for i in cluster_num:
    neigh = KNeighborsClassifier(n_neighbors=i)
    neigh = neigh.fit(X_train_norm, y_train)
    score = neigh.score(X_test_norm,y_test)
    print(f"Test accuracy(k = {i}): ",score)
```

```
Test accuracy(k = 2):  0.9297857142857143
Test accuracy(k = 10):  0.9351428571428572
Test accuracy(k = 20):  0.9270714285714285
```

I choose k = 10 and the test accuracy is 0.935.

(b) PCA

In [ ]:
```
from sklearn.decomposition import PCA
eigen_num = [2,5,10,50]
for i in eigen_num:
    pca = PCA(n_components = i)
    pca.fit(X_train_norm)
    A =  pca.transform(X_train_norm)
    A_test = pca.transform(X_test_norm)
    neigh = KNeighborsClassifier(n_neighbors=10)
    neigh.fit(A, y_train)
    score = neigh.score(A_test,y_test)
    print(f"Test accuracy for {i}D spaces",score)
```

```
Test accuracy for 2D spaces 0.33703571428571427
Test accuracy for 5D spaces 0.7499285714285714
Test accuracy for 10D spaces 0.91
Test accuracy for 50D spaces 0.9531071428571428
```

(c) Kernel PCA

For this part, the whole dataset is pretty big which can really be slow to train KPCA model. Therefore, I use Attenuated Dataset.

In [ ]:
```
data = np.load("Attenuated Dataset/pixel_data.npy")
label = np.load("Attenuated Dataset/labels.npy",allow_pickle=True)
label = np.array(list(map(int,label)))
X_train_att, X_test_att,y_train_att,y_test_att = train_test_split(data, label,test_size=0.4, random_state=42)
scaler = StandardScaler()
scaler.fit(X_train_att)
X_train_norm_att = scaler.transform(X_train_att)
X_test_norm_att = scaler.transform(X_test_att)
```

In [ ]:
```
from sklearn.decomposition import KernelPCA
from sklearn.neighbors import KNeighborsClassifier
eigen_num = [2,5,10,50]
for i in eigen_num:
    pca = KernelPCA(n_components=i, kernel='linear',gamma = 10)
    pca.fit(X_train_norm_att)
    A =  pca.transform(X_train_norm_att)
    A_test = pca.transform(X_test_norm_att)
    neigh = KNeighborsClassifier(n_neighbors=10)
    neigh.fit(A, y_train_att)
    score = neigh.score(A_test,y_test_att)
    print(f"Test accuracy for {i}D spaces",score)
```

```
Test accuracy for 2D spaces 0.3345
Test accuracy for 5D spaces 0.751
Test accuracy for 10D spaces 0.881
Test accuracy for 50D spaces 0.92175
```

(d) t-SNE

In [ ]:
```
from sklearn.manifold import TSNE
X_embedded = TSNE(n_components=2, learning_rate=800,init='random').fit_transform(pixel_data)
X_train_, X_test_,y_train_,y_test_ = train_test_split(X_embedded, labels,test_size=0.4, random_state=42)
scaler = StandardScaler()
scaler.fit(X_train_)
X_train_norm_ = scaler.transform(X_train_)
X_test_norm_ = scaler.transform(X_test_)
neigh = KNeighborsClassifier(n_neighbors=10)
neigh = neigh.fit(X_train_norm_, y_train_)
score = neigh.score(X_test_norm_,y_test_)
print("Test accuracy: ",score)
```

```
Test accuracy:  0.9706071428571429
```

(e) PCA + t-SNE

In [ ]:
```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.manifold import TSNE
from sklearn.decomposition import PCA
pca = PCA(n_components = 50)
pca.fit(X_train_norm)
A =  pca.transform(X_train_norm)
X_embedded = TSNE(n_components=2, learning_rate='auto',
                  init='random').fit_transform(A)
X_train_, X_test_,y_train_,y_test_ = train_test_split(X_embedded, y_train,test_size=0.4, random_state=42)
scaler = StandardScaler()
scaler.fit(X_train_)
X_train_norm_ = scaler.transform(X_train_)
X_test_norm_ = scaler.transform(X_test_)
neigh = KNeighborsClassifier(n_neighbors=10)
neigh = neigh.fit(X_train_norm_, y_train_)
score = neigh.score(X_test_norm_,y_test_)
print("Test accuracy: ",score)
```

```
Test accuracy:  0.9422619047619047
```