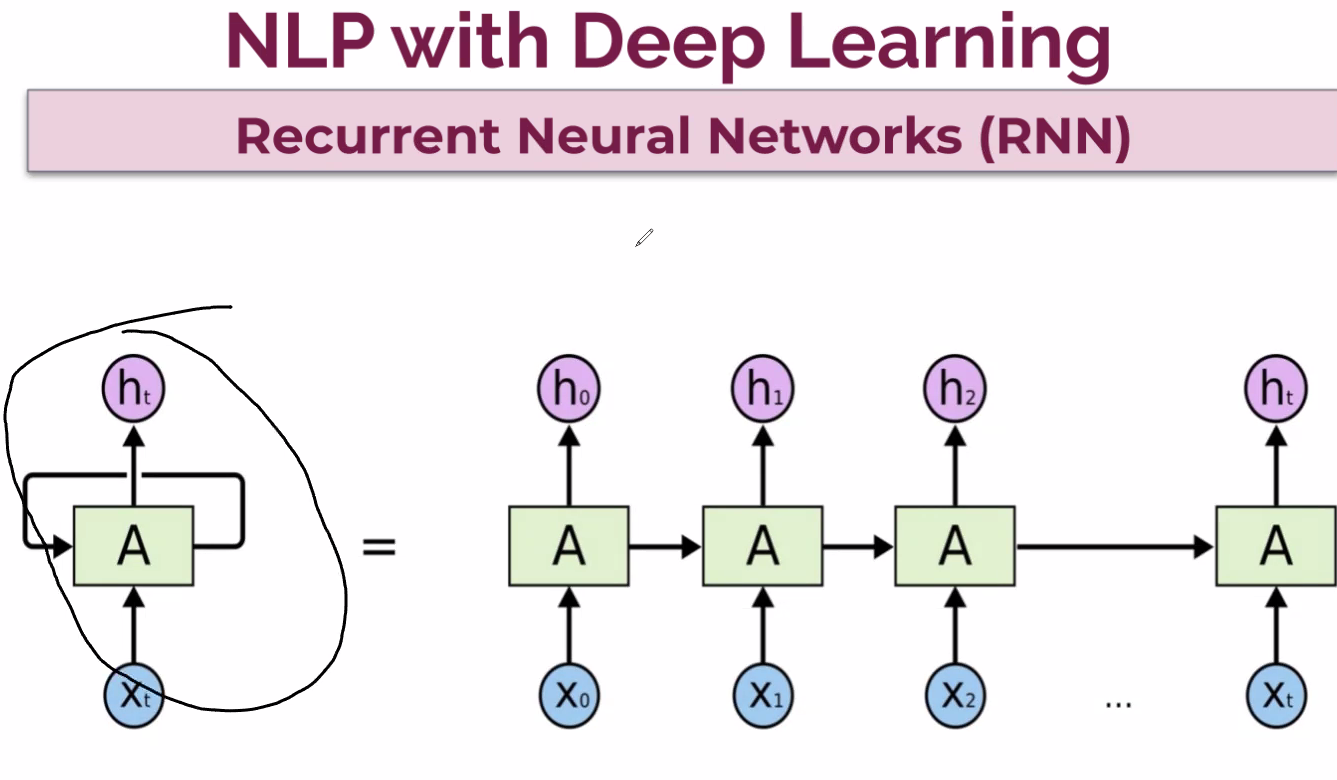
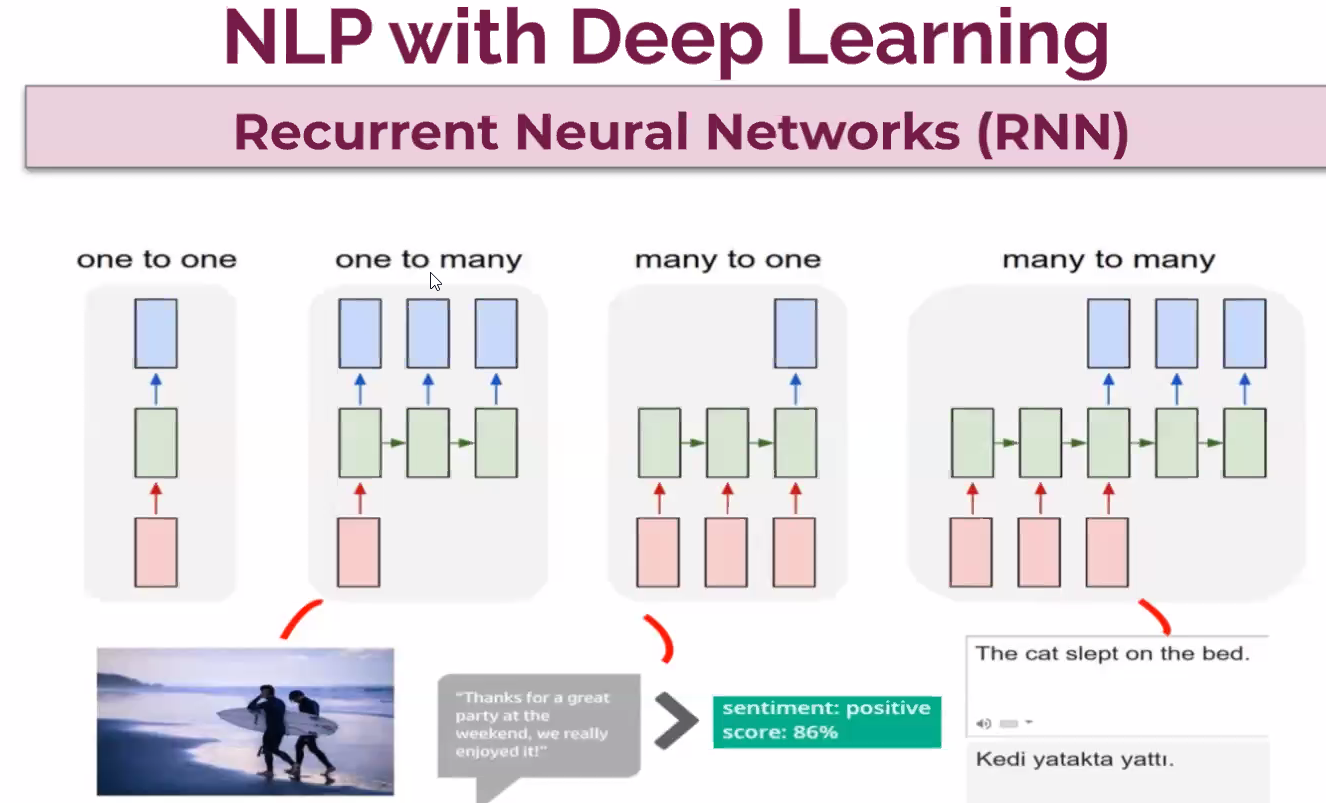


RNN ve CNN den farklı olarak her bir katman bir önceki gelen katmana bağımlı. Bir önceki layerin outputu bir diğerinin inputu oluyor. Buna katmanın hafızası diyoruz.

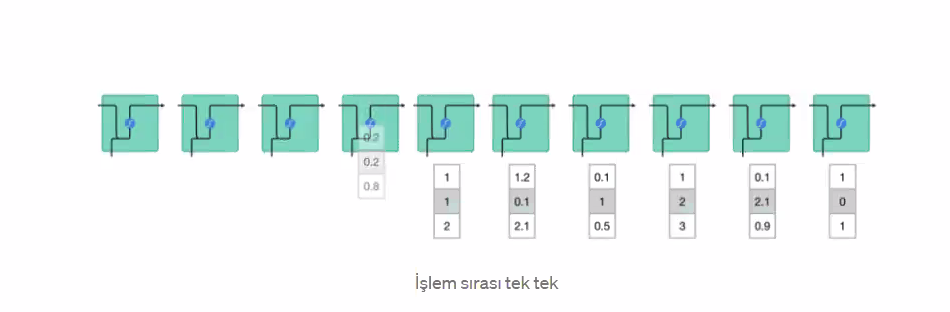


Bir cümlenin ilk kelimesininden son kelimesine kadar olan süreci ifade eder. X0 ile bir token döngüye girer ve sonrasında bu döngü diğer token ların girmesi ile devam eder. Eşitliğin sol tarafındaki döngü .izelgesinin açılmış hali sağda gösterilmektedir.

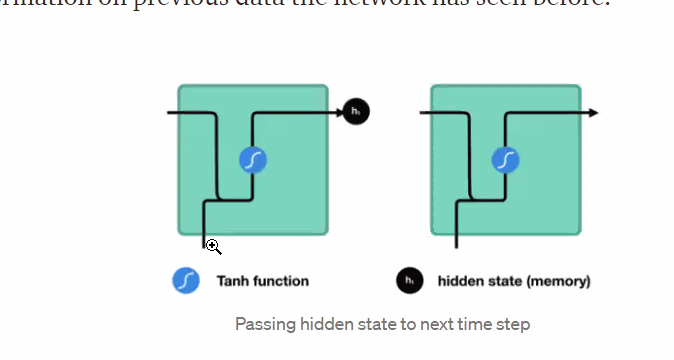
Bu döngüde genelde ilk olarak döngüye giren token unutlulmaya meyğillidir.. özellikle RNN çoğunlukla vanishin gradient e meyyaldir. Zaman zamanda exploding yaşanabilir.



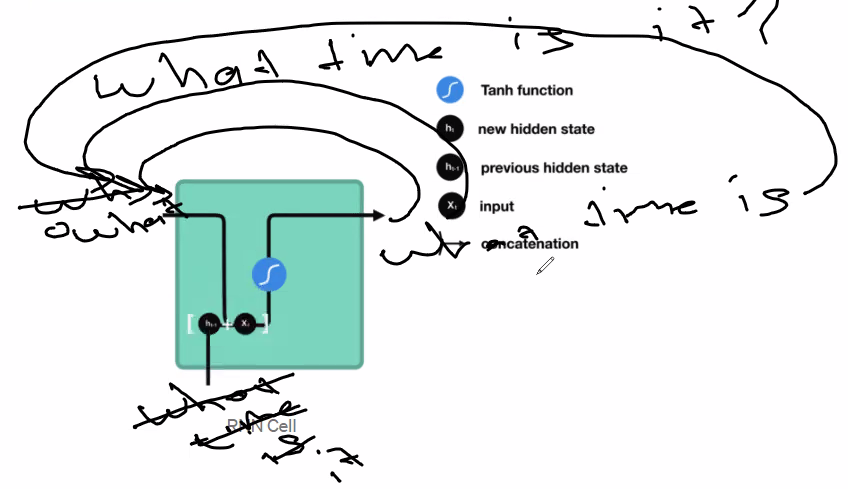
RNN modelleri bu bağlamda one to many, many to one ve many to many kullanılır.



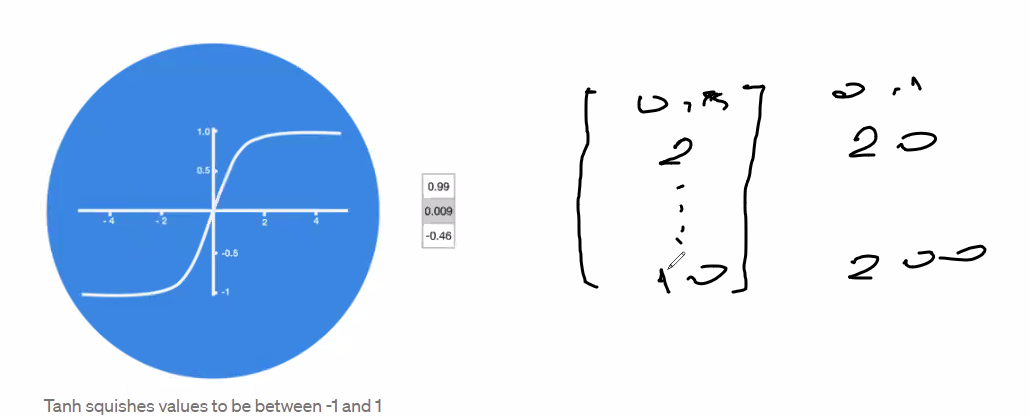
Çalışmalarda vektörlerin modele verilmesinin gözterimidir. Aslında burada sadece tke katman ve 10 token vardır. Bu bir zaman yolcuğunu ifade eder. Tokenlerin modele veriliş sırasıdır.



Burada t zamanındaki bir layer dan t+1 zamanındaki layera token geçişidir.

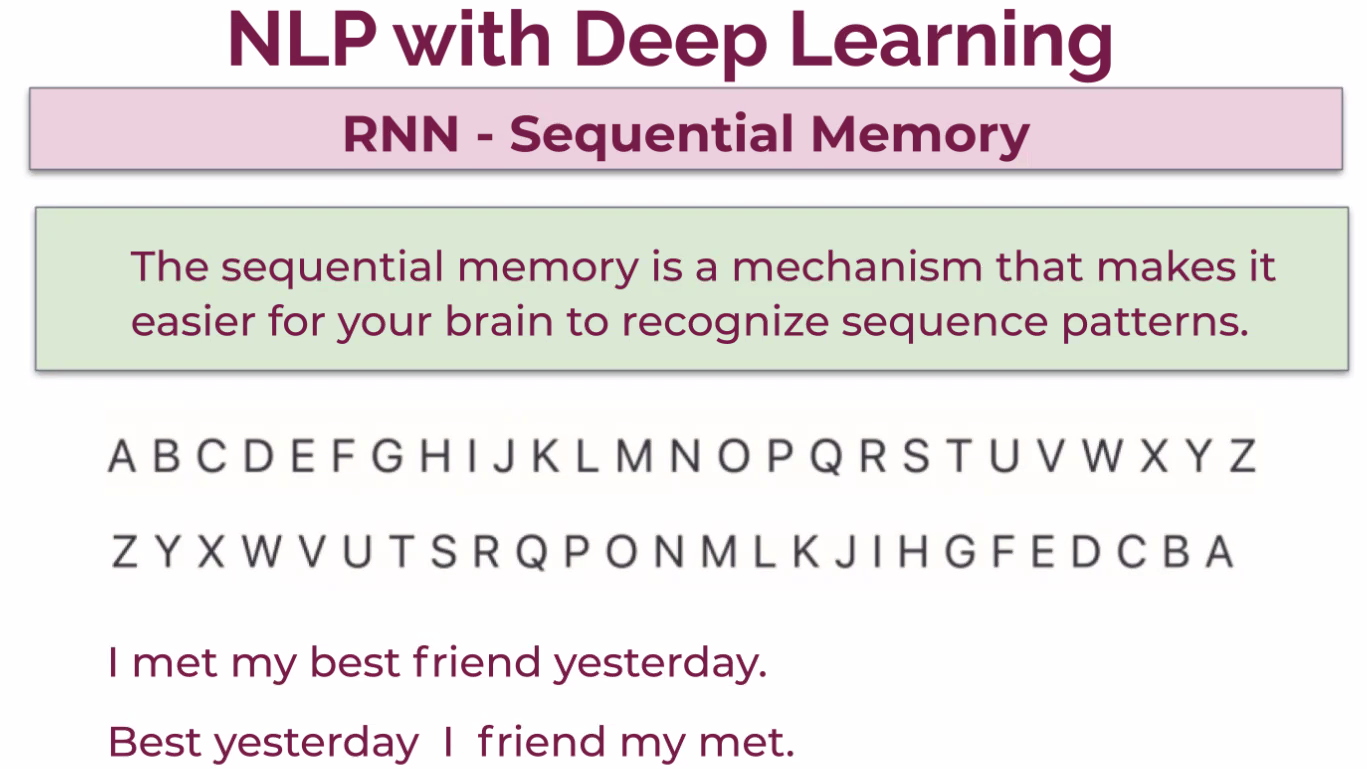


Örnekte what time is it cümlesinin RNN cell de işlenmesi gösterilmektedir.



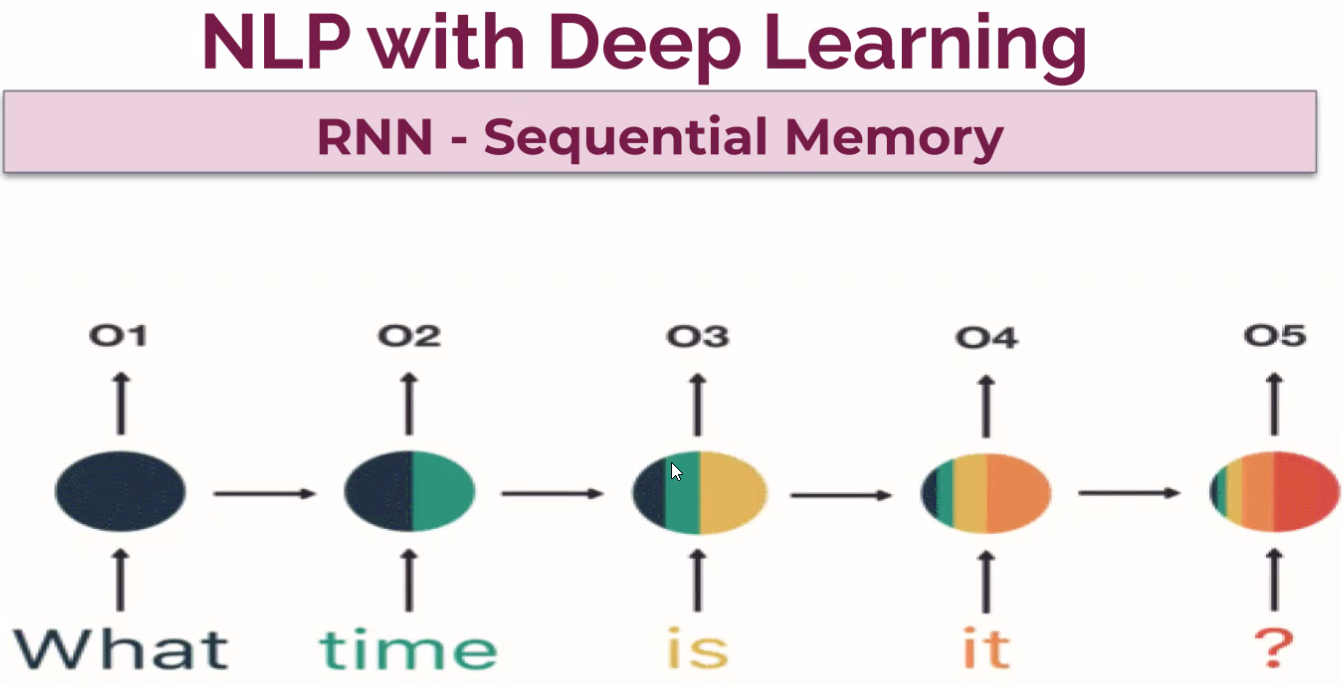
Tanh activasyonuna sokularak tokenların anlamsal ilişkisinin doğru bir şekilde olması sağlanır.

Token vektörleri -1 ile 1 arasında olmaktadır.

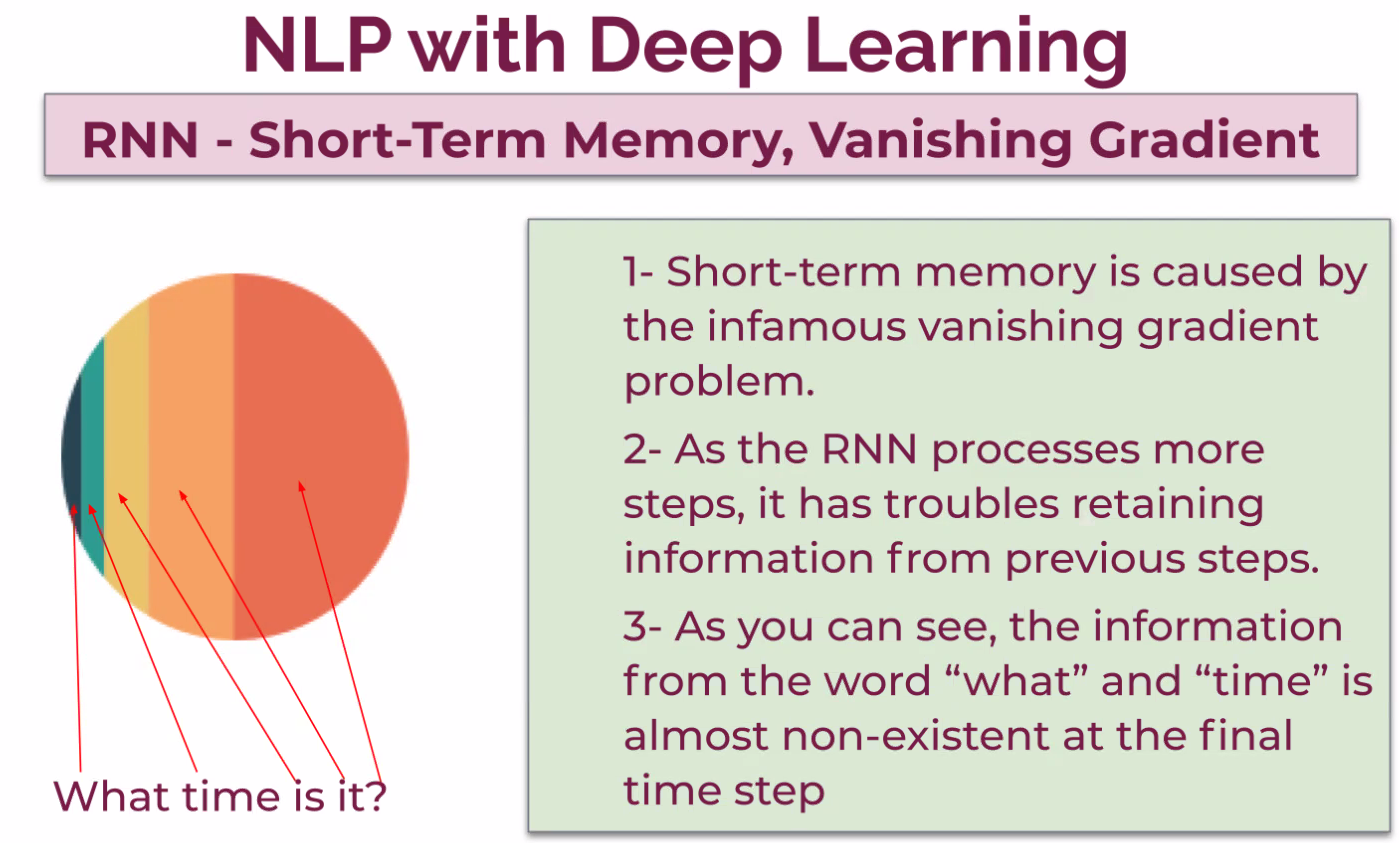


İki vektör arasındaki açı 1 ise bu kelimeler aynı demektir. Kelime uzayındaki açılarının yakın olması bu tokenların ilişkili olduğunu gösterir.

Modele verilen tokenların belirli bir dizi içinde verilmesi gerekir. Bu şekilde verilmezse model bunu anlayamaz.insanın anladığı token dizisi şeklinde modele verilmelidir.

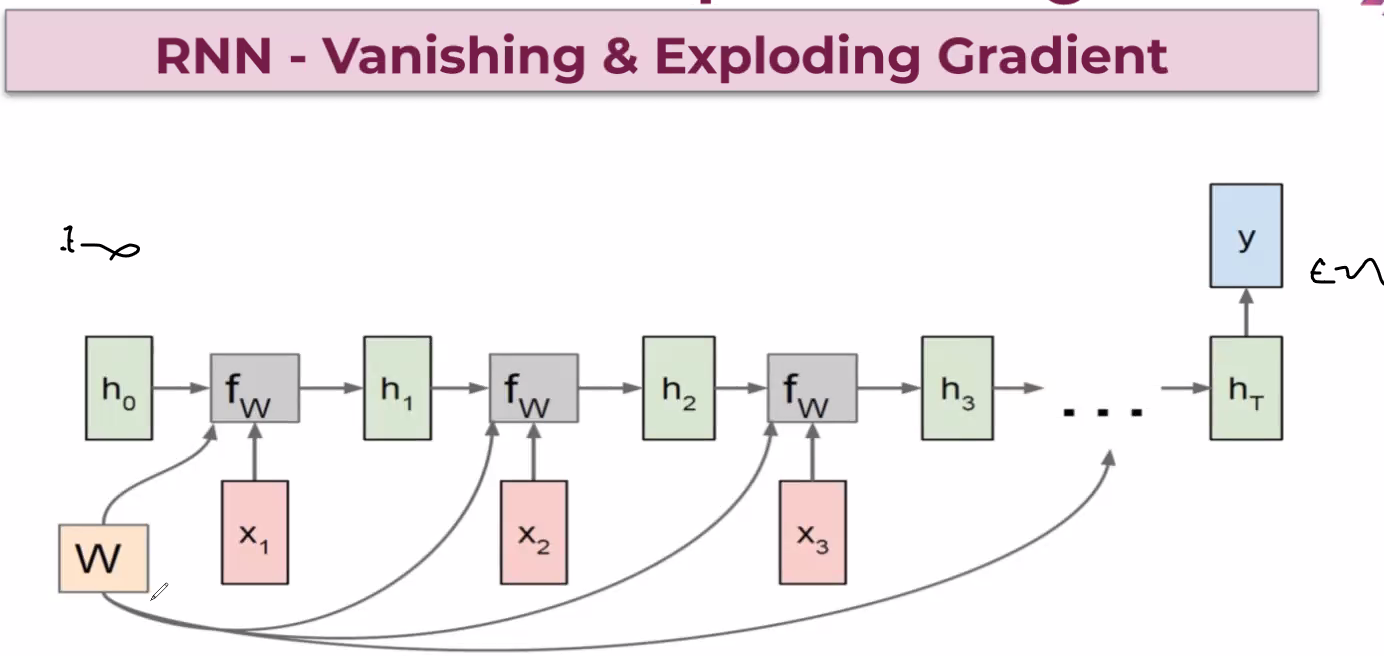


Tokenların modele girmesinde short memory sorunu vardır. Kaybolan gradyen sorunu var bu modellerde.



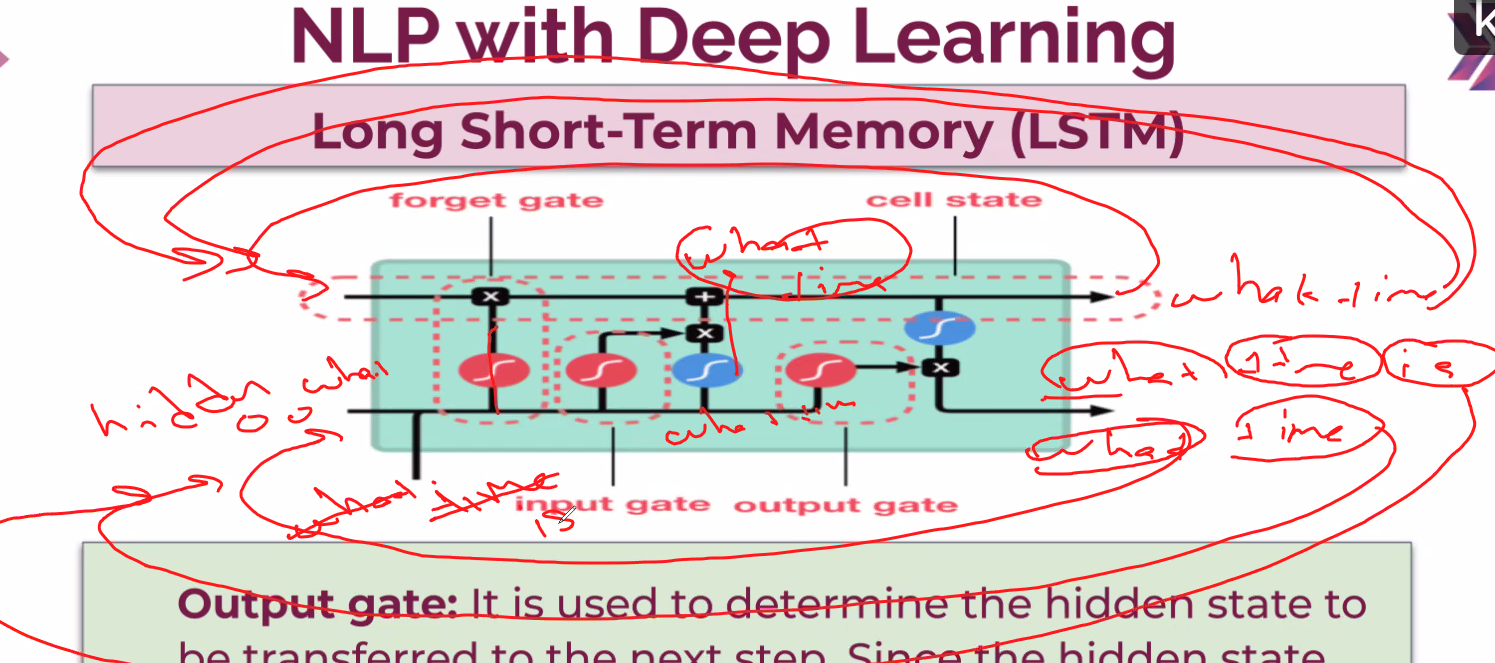


Cümlelerin fazla sayıda token olması, vanishing gradient sorununa neden olmaktadır. Sonuç olarak, ağırlık learning rate ile çarpılması ve bazk propag ile düzeltme yapılamsına rağmen, artık öğrenme meydana gelmemektedir.



Her tokende uygulanan katsayı aynı olduğu için W katsayısı 1den küçükse 4 kelimden Örn 0,02 olsun, burada be değer 4 defa birbiri ile çarpılacaktır. token sayısını çok fazla olması vanishing gradient e sebep olur.. eğer katsayı birden büyük olursa bu seferde exploding gradient meydana gelecektir. Ve öğrenme durur.

Bu problemden dolayı RNN hiç kullanılmayacak. LSTM ve GRU kullanılacak.



Hidden state RNN vardı, burda ise cell state var. Burası uzun sürelği hafıza. LSTM 3 kapısı var:

Forget gate: buradaki sigmoid fonks. İle uzun süreli hafızada (cell srate) tutulması gereken bilgilerin silinmesine ya da kalmasına karar verir.

İnput gate ile cell state token ilavesi yapıyoruz. Token Hem sigmoid den hem de tanh func dan geçiyor..

Output gate ile Hidden layer ile cell layer birleştirilir.

Cell state : uzun süreli hafıza

Hidden state :RNN deki kısa süreli hafıza.. hidden state in unutuğu tokenları cell state ona hatırlatır.

Tek desavantajı daha uzun süreli train olması.

GRU

