

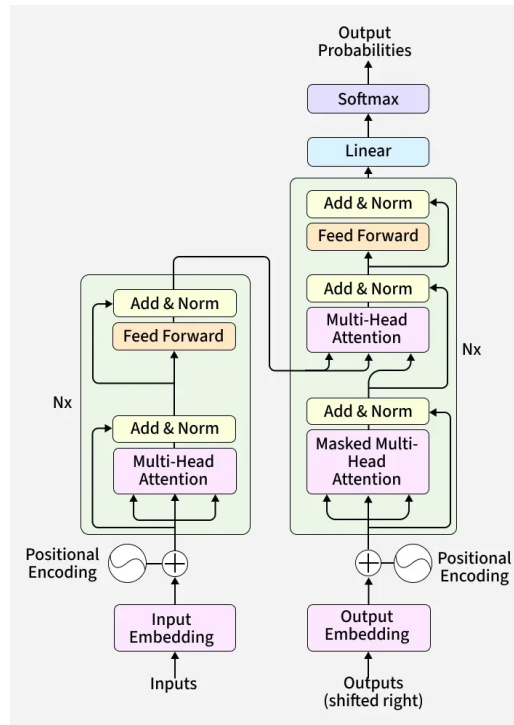


Şırnak Üniversitesi

TRANSFORMER MİMARİSİ

DERS ÖDEVİ • İleri Yapay Zekâ

Ders Sorumlusu • Doç. Dr. Yunus Eği



Şekil: Encoder-decoder Transformer mimarisinin genel görünümü.

Ödevin amacı ve kapsamı

DERS İleri Yapay Zekâ	ÖDEV TÜRÜ Grup sunumu ve teknik anlatım	ÇALIŞMA DÜZENİ 12 öğrenci • 3 grup
--------------------------	--	---------------------------------------

Ödevin temel sorusu

Bir metindeki tokenlar hangi dönüşümlerden geçerek bağlamsal vektörlere ve ardından üretilmiş bir çıktı dizisine dönüşür?

Ödev görevi

Üç grup olarak, özgün encoder-decoder Transformer mimarisinin veri akışını uçtan uca inceleyip sınıfta sununuz. Her grup, atanan bölüm için 12-15 dakikalık görsel sunum hazırlasın. Sunum; tokenization ve token ID'lerinden başlayarak embedding ile positional encoding'e, encoder self-attention ve Multi-Head Attention'dan decoder maskesi, encoder-decoder attention ve çıktı olasılıklarına kadar ilerlesin. Her aşamada girdiyi, işlemi, formülü veya örneği ve sonraki aşamaya aktarılan çıktıyı açıklayın. Tüm gruplar aynı örnek cümleyi kullanarak anlatımlarını veri akışına göre birleştirsın.

Öğrenme çıktıları

Ödev sonunda öğrencilerden şu beceriler beklenir:

- Metin tokenlarının ID ve vektör temsillerine dönüşümünü açıklamak.
- Positional Encoding'in neden ve nasıl kullanıldığını göstermek.
- Query, Key ve Value üzerinden attention ağırlıklarının oluşumunu yorumlamak.
- Encoder ve decoder alt katmanlarını doğru sırayla tanıtmak.
- Causal mask ile cross-attention'ın veri akışındaki görevini ayırt etmek.
- Bir çıktı tokenının olasılıklardan seçilip sonraki üretim adımına nasıl aktarıldığını göstermek.

Hazırlık ve sunum planı

1. Kaynağı inceleyin: Temel kaynak olarak Vaswani ve arkadaşlarının 2017 makalesini kullanın; ek kaynakları doğruluk ve açıklık için destek olarak ekleyin.
2. Aynı örneği sürdürün: Önerilen kaynak cümle “Kedi masanın üzerine çıktı.”, hedef örnek “The cat climbed onto the table.” Bu örnek tokenization’dan çıktı üretimine kadar devam etsin.
3. Veri şeklini belirtin: Her adımda girişin ne olduğunu, hangi işlemin uygulandığını ve ne çıktığını söyleyin. Gerekli yerde dizi uzunluğu, d_model, başlık sayısı veya sözlük boyutu gibi boyutları gösterin.
4. Sözlü geçiş yapın: Her grup önceki grubun çıktısını devralıp sonraki gruba anlaşılır bir cümleyle aktarsın.
5. Bireysel katkı gösterin: Dört grup üyesinin her biri kendisine ayrılan alt başlığı sunmalı ve soru-cevap bölümüne katkıda bulunmalıdır.

Sunumda beklenenler

Ödevin tek teslimi sınıfta yapılacak grup sunumudur. Ayrı rapor, özet veya yazılı ödev hazırlanmayacaktır. Grup, sunum dosyasını kullanarak 12-15 dakikada anlatım yapar. Yaklaşık 8-12 slayt önerilir. Sunum görsel açıdan zengin olmalı; görseller dekor amaçlı değil, mimariyi ve veri akışını açıklamak için kullanılmalıdır. Her grup en az beş açıklayıcı şema/görsel, aynı örnek üzerinde adım adım veri akışı, ilgili formül veya hesaplama ve son slaytta kaynakça kullanmalıdır. Her görselin ne anlattığı sözlü olarak açıklanmalıdır.

Gruplar ve ortak çalışma

Her grup bir bölümün uzmanlığını üstlenir. Grup üyeleri alt başlıkları paylaşabilir; sunum tek bir anlatı ve ortak veri örneği üzerinden ilerlemelidir.

Grup	Öğrenciler	Ana soru
1	Gülbahar Bilir — Tokenization ve token ID Bedran Gündoğan — Embedding ve vektör temsili Hüseyin Erçik — Positional Encoding ve formülü Buşra Demir — Sıra örneği, görsel akış ve grup geçişi	Metin nasıl sayısal ve sıralı temsile dönüşür?
2	Eylem Abuzeyitoğlu — Q, K, V ve scaled dot-product Hakan Bahşış — Multi-Head Attention Büyüamin Demir — Encoder bloğu ve FFN Fatma Ertürk — Residual, LayerNorm ve çıktı aktarımı	Encoder tokenlar arasında bağlamı nasıl kurar?
3	Ertan Sidar — Causal mask ve masked attention Muhammed Cüneyd Sabuncu — Decoder girdisi ve blok akışı Mahfuz İnanç — Encoder-decoder attention Mehmet Mustafa Takeş — Softmax, token seçimi ve üretim döngüsü	Decoder kaynak bilgisini kullanarak çıktıyı nasıl üretir?

Ortak örnek

Tüm gruplar tek bir örneği sürdürsün. Öneri kaynak cümle: “Kedi masanın üzerine çıktı.” Hedef örnek: “The cat climbed onto the table.” Tokenization’dan çıktı üretimine kadar aynı örnek kullanılsın.

Gruplar arası devir

- 1 → 2: “Embedding ve positional encoding eklenmiş vektör dizisi encoder’a giriyor.”
- 2 → 3: “Encoder, her kaynak token için bağlamsal vektör üretti; bunlar cross-attention’da K ve V olacak.”
- 3 → sınıf: “Decoder sonraki tokeni üretir, tahmini öneke ekler ve akışı sürdürür.”

1. Grup • Girdi temsili ve Positional Encoding

Görev: Metnin modele hangi biçimde girdiğini, kelime/alt-kelime temsillerinin nasıl vektörleştirdiğini ve konum bilgisinin nasıl eklendiğini gösterin.

Anlatılacak aşamalar

- 1. Tokenization: Cümleyi tokenlara ayırın. Token sınırlarının kelime sınırlarıyla her zaman örtüşmediğini; alt-kelime ve noktalama tokenlarının olabileceğini gösterin.
- 2. Token ID: Her tokenın sözlükteki sayısal kimliğini belirtin. ID'nin anlam vektörü olmadığını; embedding tablosundaki satırı seçtiğini açıklayın.
- 3. Embedding: Her ID'nin öğrenilebilir, aynı boyutlu vektörle eşleştiğini anlatın. Girdi [dizi uzunluğu × d_model] biçiminde bir matristir.
- 4. Sıra ihtiyacı: Self-attention tokenları birlikte işler. Aynı kelimeler farklı sırada olduğunda anlam değişebildiği için konum bilgisinin ayrıca sunulması gerektiğini örnekleyin.
- 5. Positional Encoding: Her token embedding'ine aynı boyutta bir konum vektörü eklenir. Sinüs/kosinüs formülündeki pos, i ve d_model terimlerini açıklayın. Farklı frekanslar konum örüntülerini farklı ölçeklerde temsil eder.
- 6. Mimari not: Özgün makalede positional encoding giriş embedding'lerine eklenir. Bu ödevde sinusoidal yöntemi anlatın; modern modellerde farklı konum yöntemleri bulunduğunu kısa not olarak ekleyin.
- 7. Devir: Grup 2'ye konum bilgili vektör dizisini ve encoder'ın ilk işleminin Self-Attention olduğunu gösterin.

Sinüzoidal Positional Encoding denklemleri

$$PE_{(pos, 2i)} = \sin\left(\frac{pos}{10000^{2i/d_{model}}}\right)$$

$$PE_{(pos, 2i+1)} = \cos\left(\frac{pos}{10000^{2i/d_{model}}}\right)$$

Burada pos tokenın konumudur, i vektör boyut çiftini, d_model ise embedding boyutunu belirtir.

Giriş temsili

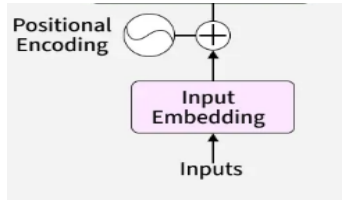
$$\mathbf{x}_{pos} = \mathbf{e}_{token} + \mathbf{p}_{pos}$$

Hazırlanacak görseller

Metin → tokenlar → ID → embedding → konum vektörü ekleme şeması; iki farklı sıralamanın karşılaştırması; temsili vektör toplamı; tensor boyutlarının küçük bir örneği.

Tartışma soruları

Token ID neden doğrudan anlamı temsil etmez? Positional Encoding olmazsa model hangi bilgiyi kaybedebilir? Her aşamada verinin biçimi nedir?



Görsel: Input Embedding'e Positional Encoding eklenmesi.

2. Grup • Encoder, Self-Attention ve Multi-Head Attention

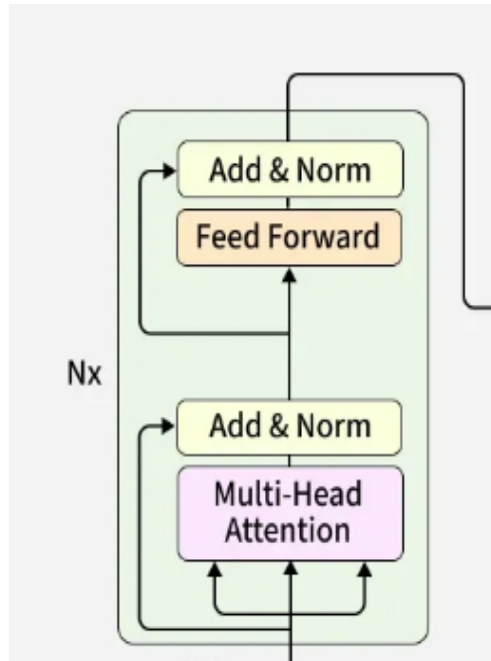
Görev: Encoder'ın dizideki ilişkileri hesaba katarak token başına nasıl bağlamsal temsiller ürettiğini hesaplama akışıyla açıklayın.

Attention hesaplama şeması

Sunumda attention işlemini Q, K ve V akışını izleyerek kendi hazırladığınız bir şema üzerinde anlatın.

Anlatılacak aşamalar

- 1. Encoder girdisi: Grup 1'den gelen X vektör dizisini alın. Her vektör token içeriği ve konum bilgisini taşır. Encoder çıktısı, yine token sayısını koruyan bağlamsal vektör dizisidir.
- 2. Q, K, V: X, öğrenilen WQ, WK, WV matrisleriyle Query, Key, Value temsillerine dönüştürülür. Sezgisel olarak Q "ne arıyorum?", K "hangi eşleşme var?", V "hangi bilgiyi taşıyorum?"
- 3. Skorlar: QK^T ile her sorgu tokenının her key tokenına benzerliği hesaplanır. $\sqrt{d_k}$ ölçeklemesi skorların büyümesini dengeler. Softmax her sorgu için ağırlıkları normalize eder.
- 4. Ağırlıklı bilgi: Softmax ağırlıkları V'leri birleştirir. Her token, ilişkili diğer tokenlardan gelen bilgiyi ağırlıklı toplayarak yeni temsili üretir. Encoder self-attention'da tokenlar tüm kaynak konumlarına bakabilir; padding maskesi ayrı ele alınabilir.
- 5. Multi-Head: Dönüşüm birden fazla başlıkta farklı öğrenilmiş projeksiyonlarla paralel yapılır. Başlık çıktıları birleştirilir ve doğrusal projeksiyondan geçirilir. Başlıkların her zaman yorumlanabilir, sabit anlamları olduğu iddia edilmemelidir.
- 6. Encoder bloğu: Multi-Head Self-Attention → residual + LayerNorm → position-wise FFN → residual + LayerNorm. FFN her konumda aynı ağla uygulanır; tokenlar arası bilgi alışverişi attention'da olur.
- 7. Tekrarlama ve aktarım: Blok N kez yığılır. Son katman her kaynak token için bağlamsal çıktı verir; bu dizi cross-attention'da K ve V olarak kullanılır.



Görsel: Multi-Head Attention, Add & Norm ve Feed-Forward alt katmanlarının encoder bloğundaki sırası.

Encoder bloğu ve attention denklemleri

Encoder'daki attention alt katmanından gelen çıktı, residual bağlantıyla alt katman girdisine eklenir ve Layer Normalization uygulanır. Ardından her token konumuna aynı Feed-Forward ağı uygulanır; ikinci residual + normalization adımı encoder çıktısını oluşturur.

Scaled dot-product ve Multi-Head formülleri

Scaled dot-product attention

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Multi-Head Attention

$$\text{head}_j = \text{Attention}(QW_j^Q, KW_j^K, VW_j^V)$$

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_h)W^O$$

d_k , key vektörünün boyutudur. QK benzerlik skorlarını, $\sqrt{d_k}$ ölçeklemeyi, softmax ise her sorgu için normalize edilmiş attention ağırlıklarını verir. Value vektörleri bu ağırlıklarla birleştirilir. Başlıklar birleştirildikten sonra W^O doğrusal dönüşümü uygulanır.

Tek bir sorgu tokeni için temsili örnek

Bir sorgu tokeninin üç key tokeni için ölçeklenmiş attention skorlarının sırasıyla 2, 1 ve 0 olduğunu varsayalım. Softmax yaklaşık 0.67, 0.24 ve 0.09 ağırlıklarını üretir. Çıktı, bu ağırlıklarla V vektörlerinin ağırlıklı toplamıdır: $0.67V_1 + 0.24V_2 + 0.09V_3$. Sayılar yalnızca işlem sırasını göstermek için temsili seçilmiştir.

Öğrencilerin bu örnekte açıklaması gerekenler

En yüksek skoru alan key'in ağırlığı neden daha büyüktür? Ağırlıkların toplamı yaklaşık neden 1 eder? Bir key'in skoru düşerse çıktıdaki ilgili Value katkısı nasıl değişir?

3. Grup • Decoder, mask ve çıktı üretimi

Görev: Decoder'ın önceki hedef tokenları ve encoder kaynak temsillerini kullanarak sonraki tokenı nasıl ürettiğini; eğitim ve çıkarım süreçlerini ayrı ayrı anlatın.

Decoder akışını kendi sunum görseliniz üzerinde gösterin: masked self-attention, encoder-decoder attention, FFN ve çıktı katmanı sırasını; encoder çıktısından K/V aktarımını ayrıca işaretleyin.

Anlatılacak aşamalar

- 1. Decoder girdisi: Eğitimde hedef dizi sağa kaydırılır. Başlangıç tokeni ile başlayan decoder her konumda sonraki hedef tokenı tahmin eder. Örnek: giriş "<bos> The cat", hedef "The cat climbed". Teacher forcing'i tanımlayın.
- 2. Masked Self-Attention: Decoder t konumunda yalnız $\leq t$ konumlarını görür. Gelecek skorlarına $-\infty$ benzeri maske uygulanır; softmax ağırlığı sıfırlar. Üç token için nedensel görünürlük matrisini çizin.
- 3. Encoder-Decoder Attention: Decoder temsilleri Q; encoder son çıktıları K ve V olur. Böylece decoder çıktının mevcut noktasında kaynak cümledeki ilgili tokenlarına odaklanabilir. Bağlantı okunu şemada özellikle gösterin.
- 4. Decoder bloğu: Masked MHA $\rightarrow$ Add & Norm $\rightarrow$ cross-attention $\rightarrow$ Add & Norm $\rightarrow$ FFN $\rightarrow$ Add & Norm. Bu alt katmanlar decoder katmanlarında tekrarlanır.
- 5. Olasılık: Son decoder temsili Linear katmanla sözlük boyutundaki logitlere, softmax ile olasılık dağılımına dönüşür. Argmax, örnekleme ve beam search yöntemlerini kısaca tanıttın.
- 6. Otoregresif üretim: Seçilen token öneke eklenir; decoder yeniden çalıştırılır ve sıradaki token tahmin edilir. Süreç bitiş tokeni (<eos>) veya sınır koşuluna kadar sürer.
- 7. Eğitim/çıkartım: Eğitimde doğru hedef öneki maskeyle paralel işlenebilir; çıkartımda önceki tahminler sonraki adımın girdisidir ve üretim adım adım ilerler.

Decoder çıktı olasılığı

$$\text{logits}_t = W_o \mathbf{h}_t + \mathbf{b}_o$$

$$p(y_t | y_{<t}, x) = \text{softmax}(\text{logits}_t)$$

$\mathbf{h}_t$ decoder'ın son katmandaki gizil temsildir; W_o ve $\mathbf{b}_o$ sözlükteki tokenlara ait puanları üretir. Softmax bu puanları olasılık dağılımına dönüştürür.

Hazırlanacak görseller

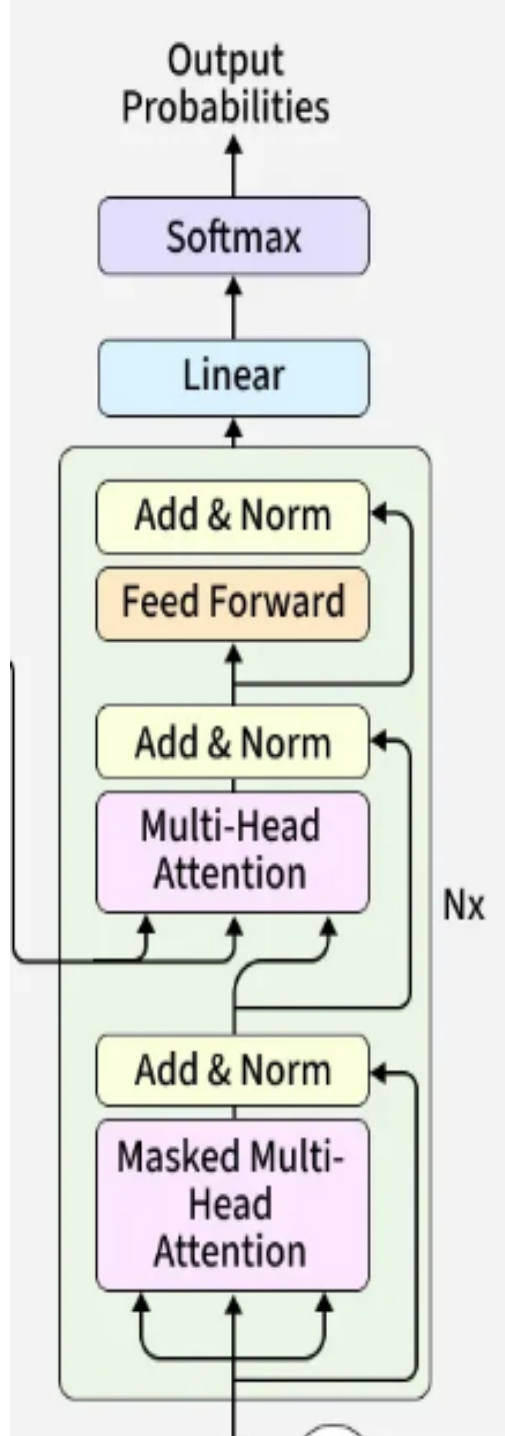
Decoder blok şeması; üç-token causal mask matrisi; encoder çıkışından cross-attention'a K/V oku; Linear $\rightarrow$ Softmax $\rightarrow$ token seçimi; adım adım token üretim çizelgesi.

Tartışma soruları

Mask olmazsa decoder hangi bilgiyi görebilir? Cross-attention'da Q nereden, K/V nereden gelir? Eğitim ve çıkartımda hedef öneki nasıl değişir?

Decoder katmanları ve çıktı akışı

Bu görselde masked self-attention, encoder-decoder attention, Feed-Forward alt katmanı ve çıktı olasılıklarına uzanan Linear-Softmax aşamaları görülmektedir. Şemayı incelerken encoder çıktıların cross-attention'a bağlandığı bağlantıyı da açıklayın.



Görsel: Decoder yığını ve çıktı katmanı. Kullanıcı tarafından sağlanmıştır.

Ortak prova, akış kontrolü ve değerlendirme

Her grup kendisinden önceki grubun bıraktığı temsille başlamalıdır. “Sonra attention gelir” demek yerine girdinin şeklini, işlemi ve çıkan temsili belirtin.

Aşama	Veri / temsil	Sonraki adıma aktarılan
Tokenization	Token dizisi ve ID'ler	Embedding tablosundan vektörler
Embedding + konum	X : dizi uzunluğu $\times$ d_{model}	Encoder girdi dizisi
Encoder	Kaynak token başına bağlamsal vektör	Cross-attention K ve V
Decoder masked attention	Hedef önekenden nedensel temsiller	Cross-attention Q
Çıkış katmanı	Sözlük boyutunda logit/olasılık	Seçilen sonraki token

Ortak tartışma

- Hangi işlemlerde token sayısı korunur, hangi aşamada sözlük olasılıklarına geçilir?
- Positional Encoding sıraya dair hangi ek bilgiyi taşır?
- Self-attention ve cross-attention'da Q/K/V kaynakları nasıl farklıdır?
- Residual bağlantı hangi sinyalleri toplar; LayerNorm attention'ın yerine mi geçer?
- Causal mask ile padding mask aynı işi mi yapar?
- Encoder-decoder Transformer ile decoder-only mimari hangi noktada ayrılır?

Teslim ve sunum standardı

Her grup 12-15 dakika sunar; tüm üyeler kendi alt konularını anlatır ve soru-cevap bölümüne katılır. En az beş açıklayıcı şema/görsel kullanılır. Her formüldeki semboller, verinin aşamalar arasındaki dönüşümü ve görseldeki oklar açıklanır. Sunumda ortak kaynak-hedef örneği, veri/tensor akışını gösteren bir slayt, gruplar arası geçiş ve encoder çıktısından decoder cross-attention'a uzanan bağlantı bulunmalıdır. Kaynakça son slaytta yer alır; sunum sonunda iki tartışma sorusu yöneltilir. Rapor veya özet teslimi yoktur.

Değerlendirme

Ölçüt	Ağırlık
Mimari ve veri akışının doğruluğu	30%
Ayrıntı ve anlatım açıklığı	25%
Görsel/şema ve örneğin öğreticiliği	20%
Gruplar arası bütünlük	15%
Kaynak kullanımı ve sunum düzeni	10%

Kaynaklar

Teknik mimari ve attention denklemleri özgün encoder-decoder Transformer tanımına dayanır. Sunumda kullanılacak görseller öğrenciler tarafından hazırlanmalı veya dış kaynaktan alındıysa kaynakları açıkça gösterilmelidir.

Temel kaynak

Vaswani, A., et al. (2017). “Attention Is All You Need.” Advances in Neural Information Processing Systems 30 (NeurIPS 2017). Özgün makale; encoder/decoder blokları, scaled dot-product attention, Multi-Head Attention, positional encoding ve masked decoder akışını tanımlar.

<https://papers.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf>
<https://arxiv.org/abs/1706.03762>

Ek kaynaklar

Google Research. “Attention Is All You Need.” Araştırma yayını ve Transformer’ın attention tabanlı encoder-decoder mimarisi hakkında özet.

<https://research.google/pubs/attention-is-all-you-need/>

Hugging Face Course. “How do Transformers work?” Attention maskeleri ve Transformer aileleri hakkında destekleyici ders içeriği.

<https://huggingface.co/docs/course/main/en/chapter1/4>

Görsel kullanımı

Öğrenciler sunum görsellerini kendileri hazırlayabilir veya güvenilir kaynaklardan seçebilir. Her dış kaynaktan alınan görselin kaynağı slaytta ya da kaynakçada belirtilmelidir. Bu yönerge, görseller sonradan eklenebilecek şekilde metin odaklı hazırlanmıştır.