

```
pub struct Nemotron { /* private fields */ }
```

Nemotron streaming ASR model (0.6B parameters). We dont apply mel normalization unlike others...

## Implementations

### impl Nemotron

```
pub fn from_pretrained<P: AsRef<Path>>(<br>    path: P,<br>    exec_config: Option<ExecutionConfig>,<br>) -> Result<Self>
```

Load Nemotron model from directory.

Required files:

- encoder.onnx + encoder.onnx.data
- decoder\_joint.onnx
- tokenizer.model

### Examples found in repository

```
152         duration / elapsed.as_secs_f32()<br>153     );<br>154<br>155     Ok(())<br>156 }
```

examples/streaming.rs (line 116)

```
pub fn reset(&mut self)
```

Reset all state for new utterance

```
pub fn get_transcript(&self) -> String
```

Get the full accumulated transcript

[Examples found in repository](#) (?)

```
152         duration / elapsed.as_secs_f32()
153     );
154
155     Ok(())
156 }
```

[examples/streaming.rs \(line 145\)](#)

```
pub fn transcribe_file<P: AsRef<Path>>(<
    &mut self,
    audio_path: P,
> -> Result<String>
```

note that, offline transcription for testing/debugging and for some curious ppl :-). with following function too (transcribe\_audio)

```
pub fn transcribe_audio(&mut self, audio: &[f32]) -> Result<String>
```

Transcribe audio samples (non-streaming)

```
pub fn transcribe_chunk(&mut self, audio_chunk: &[f32]) -> Result<String>
```

Stream transcribe a chunk of audio (call repeatedly for real-time).

This buffers raw audio and computes mel spectrograms over the full buffer to avoid edge effects at chunk boundaries.

[Examples found in repository](#) (?)

```
152         duration / elapsed.as_secs_f32()
153     );
154
155     Ok(())
156 }
```

[examples/streaming.rs \(line 130\)](#)

## Auto Trait Implementations

---

```
impl Freeze for Nemotron
```

```
impl !RefUnwindSafe for Nemotron
```

```
impl Send for Nemotron
impl Sync for Nemotron
impl Unpin for Nemotron
impl !UnwindSafe for Nemotron
```

## Blanket Implementations

---

```
impl<T> Any for T
where
    T: 'static + ?Sized,

impl<T> Borrow<T> for T
where
    T: ?Sized,

impl<T> BorrowMut<T> for T
where
    T: ?Sized,

impl<T> From<T> for T

impl<T> Instrument for T

impl<T, U> Into<U> for T
where
    U: From<T>,

impl<T> IntoEither for T

impl<T> Pointable for T

impl<T, U> TryFrom<U> for T
where
    U: Into<T>,

impl<T, U> TryInto<U> for T
where
    U: TryFrom<T>,

impl<V, T> VZip<V> for T
where
    V: MultiLane<T>,

impl<T> WithSubscriber for T
```

